

short_1

November 11, 2023

1 Install necessary packages

```
[1]: !pip install autogluon matplotlib
```

```
Requirement already satisfied: autogluon in /opt/conda/lib/python3.10/site-  
packages (0.8.2)  
Requirement already satisfied: matplotlib in /opt/conda/lib/python3.10/site-  
packages (3.6.3)  
Requirement already satisfied: autogluon.core==0.8.2 in  
/opt/conda/lib/python3.10/site-packages (from  
autogluon.core[all]==0.8.2->autogluon) (0.8.2)  
Requirement already satisfied: autogluon.features==0.8.2 in  
/opt/conda/lib/python3.10/site-packages (from autogluon) (0.8.2)  
Requirement already satisfied: autogluon.tabular==0.8.2 in  
/opt/conda/lib/python3.10/site-packages (from  
autogluon.tabular[all]==0.8.2->autogluon) (0.8.2)  
Requirement already satisfied: autogluon.multimodal==0.8.2 in  
/opt/conda/lib/python3.10/site-packages (from autogluon) (0.8.2)  
Requirement already satisfied: autogluon.timeseries==0.8.2 in  
/opt/conda/lib/python3.10/site-packages (from  
autogluon.timeseries[all]==0.8.2->autogluon) (0.8.2)  
Requirement already satisfied: numpy<1.27,>=1.21 in  
/opt/conda/lib/python3.10/site-packages (from  
autogluon.core==0.8.2->autogluon.core[all]==0.8.2->autogluon) (1.23.5)  
Requirement already satisfied: scipy<1.12,>=1.5.4 in  
/opt/conda/lib/python3.10/site-packages (from  
autogluon.core==0.8.2->autogluon.core[all]==0.8.2->autogluon) (1.11.2)  
Requirement already satisfied: scikit-learn<1.3,>=1.0 in  
/opt/conda/lib/python3.10/site-packages (from  
autogluon.core==0.8.2->autogluon.core[all]==0.8.2->autogluon) (1.2.2)  
Requirement already satisfied: networkx<4,>=3.0 in  
/opt/conda/lib/python3.10/site-packages (from  
autogluon.core==0.8.2->autogluon.core[all]==0.8.2->autogluon) (3.1)  
Requirement already satisfied: pandas<1.6,>=1.4.1 in  
/opt/conda/lib/python3.10/site-packages (from  
autogluon.core==0.8.2->autogluon.core[all]==0.8.2->autogluon) (1.5.3)  
Requirement already satisfied: tqdm<5,>=4.38 in /opt/conda/lib/python3.10/site-  
packages (from autogluon.core==0.8.2->autogluon.core[all]==0.8.2->autogluon)
```

(4.65.2)
Requirement already satisfied: requests in /opt/conda/lib/python3.10/site-packages (from autogluon.core==0.8.2->autogluon.core[all]==0.8.2->autogluon) (2.28.2)
Requirement already satisfied: boto3<2,>=1.10 in /opt/conda/lib/python3.10/site-packages (from autogluon.core==0.8.2->autogluon.core[all]==0.8.2->autogluon) (1.28.68)
Requirement already satisfied: autogluon.common==0.8.2 in /opt/conda/lib/python3.10/site-packages (from autogluon.core==0.8.2->autogluon.core[all]==0.8.2->autogluon) (0.8.2)
Requirement already satisfied: hyperopt<0.2.8,>=0.2.7 in /opt/conda/lib/python3.10/site-packages (from autogluon.core[all]==0.8.2->autogluon) (0.2.7)
Requirement already satisfied: ray<2.4,>=2.3 in /opt/conda/lib/python3.10/site-packages (from ray[default]<2.4,>=2.3; extra == "all"->autogluon.core[all]==0.8.2->autogluon) (2.3.1)
Requirement already satisfied: pydantic<2.0,>=1.10.4 in /opt/conda/lib/python3.10/site-packages (from autogluon.core[all]==0.8.2->autogluon) (1.10.12)
Requirement already satisfied: grpcio<=1.50.0,>=1.42.0 in /opt/conda/lib/python3.10/site-packages (from autogluon.core[all]==0.8.2->autogluon) (1.50.0)
Requirement already satisfied: Pillow<9.6,>=9.3 in /opt/conda/lib/python3.10/site-packages (from autogluon.multimodal==0.8.2->autogluon) (9.5.0)
Requirement already satisfied: jsonschema<4.18,>=4.14 in /opt/conda/lib/python3.10/site-packages (from autogluon.multimodal==0.8.2->autogluon) (4.17.3)
Requirement already satisfied: sequeval<1.3.0,>=1.2.2 in /opt/conda/lib/python3.10/site-packages (from autogluon.multimodal==0.8.2->autogluon) (1.2.2)
Requirement already satisfied: evaluate<0.4.0,>=0.2.2 in /opt/conda/lib/python3.10/site-packages (from autogluon.multimodal==0.8.2->autogluon) (0.3.0)
Requirement already satisfied: accelerate<0.17,>=0.9 in /opt/conda/lib/python3.10/site-packages (from autogluon.multimodal==0.8.2->autogluon) (0.16.0)
Requirement already satisfied: timm<0.10.0,>=0.9.2 in /opt/conda/lib/python3.10/site-packages (from autogluon.multimodal==0.8.2->autogluon) (0.9.8)
Requirement already satisfied: torch<1.14,>=1.9 in /opt/conda/lib/python3.10/site-packages (from autogluon.multimodal==0.8.2->autogluon) (1.13.1+cu117)
Requirement already satisfied: torchvision<0.15.0 in /opt/conda/lib/python3.10/site-packages (from autogluon.multimodal==0.8.2->autogluon) (0.14.1+cu117)
Requirement already satisfied: scikit-image<0.20.0,>=0.19.1 in /opt/conda/lib/python3.10/site-packages (from

```

autogluon.multimodal==0.8.2->autogluon) (0.19.3)
Requirement already satisfied: pytorch-lightning<1.10.0,>=1.9.0 in
/opt/conda/lib/python3.10/site-packages (from
autogluon.multimodal==0.8.2->autogluon) (1.9.5)
Requirement already satisfied: text-unidecode<1.4,>=1.3 in
/opt/conda/lib/python3.10/site-packages (from
autogluon.multimodal==0.8.2->autogluon) (1.3)
Requirement already satisfied: torchmetrics<0.12.0,>=0.11.0 in
/opt/conda/lib/python3.10/site-packages (from
autogluon.multimodal==0.8.2->autogluon) (0.11.4)
Requirement already satisfied: transformers<4.27.0,>=4.23.0 in
/opt/conda/lib/python3.10/site-packages (from transformers[sentencepiece]<4.27.0
,>=4.23.0->autogluon.multimodal==0.8.2->autogluon) (4.26.1)
Requirement already satisfied: nptyping<2.5.0,>=1.4.4 in
/opt/conda/lib/python3.10/site-packages (from
autogluon.multimodal==0.8.2->autogluon) (2.4.1)
Requirement already satisfied: omegaconf<2.3.0,>=2.1.1 in
/opt/conda/lib/python3.10/site-packages (from
autogluon.multimodal==0.8.2->autogluon) (2.2.3)
Requirement already satisfied: pytorch-metric-learning<2.0,>=1.3.0 in
/opt/conda/lib/python3.10/site-packages (from
autogluon.multimodal==0.8.2->autogluon) (1.7.3)
Requirement already satisfied: nlpaug<1.2.0,>=1.1.10 in
/opt/conda/lib/python3.10/site-packages (from
autogluon.multimodal==0.8.2->autogluon) (1.1.11)
Requirement already satisfied: nltk<4.0.0,>=3.4.5 in
/opt/conda/lib/python3.10/site-packages (from
autogluon.multimodal==0.8.2->autogluon) (3.8.1)
Requirement already satisfied: openmim<0.4.0,>=0.3.7 in
/opt/conda/lib/python3.10/site-packages (from
autogluon.multimodal==0.8.2->autogluon) (0.3.9)
Requirement already satisfied: defusedxml<0.7.2,>=0.7.1 in
/opt/conda/lib/python3.10/site-packages (from
autogluon.multimodal==0.8.2->autogluon) (0.7.1)
Requirement already satisfied: jinja2<3.2,>=3.0.3 in
/opt/conda/lib/python3.10/site-packages (from
autogluon.multimodal==0.8.2->autogluon) (3.1.2)
Requirement already satisfied: tensorboard<3,>=2.9 in
/opt/conda/lib/python3.10/site-packages (from
autogluon.multimodal==0.8.2->autogluon) (2.15.0)
Requirement already satisfied: pytesseract<0.3.11,>=0.3.9 in
/opt/conda/lib/python3.10/site-packages (from
autogluon.multimodal==0.8.2->autogluon) (0.3.10)
Requirement already satisfied: lightgbm<3.4,>=3.3 in
/opt/conda/lib/python3.10/site-packages (from
autogluon.tabular[all]==0.8.2->autogluon) (3.3.5)
Requirement already satisfied: xgboost<1.8,>=1.6 in
/opt/conda/lib/python3.10/site-packages (from

```

autogluon.tabular[all]==0.8.2->autogluon) (1.7.6)
 Requirement already satisfied: fastai<2.8,>=2.3.1 in
 /opt/conda/lib/python3.10/site-packages (from
 autogluon.tabular[all]==0.8.2->autogluon) (2.7.13)
 Requirement already satisfied: catboost<1.3,>=1.1 in
 /opt/conda/lib/python3.10/site-packages (from
 autogluon.tabular[all]==0.8.2->autogluon) (1.2.2)
 Requirement already satisfied: joblib<2,>=1.1 in /opt/conda/lib/python3.10/site-
 packages (from
 autogluon.timeseries==0.8.2->autogluon.timeseries[all]==0.8.2->autogluon)
 (1.3.2)
 Requirement already satisfied: statsmodels<0.15,>=0.13.0 in
 /opt/conda/lib/python3.10/site-packages (from
 autogluon.timeseries==0.8.2->autogluon.timeseries[all]==0.8.2->autogluon)
 (0.14.0)
 Requirement already satisfied: gluonts<0.14,>=0.13.1 in
 /opt/conda/lib/python3.10/site-packages (from
 autogluon.timeseries==0.8.2->autogluon.timeseries[all]==0.8.2->autogluon)
 (0.13.7)
 Requirement already satisfied: statsforecast<1.5,>=1.4.0 in
 /opt/conda/lib/python3.10/site-packages (from
 autogluon.timeseries==0.8.2->autogluon.timeseries[all]==0.8.2->autogluon)
 (1.4.0)
 Requirement already satisfied: mlforecast<0.7.4,>=0.7.0 in
 /opt/conda/lib/python3.10/site-packages (from
 autogluon.timeseries==0.8.2->autogluon.timeseries[all]==0.8.2->autogluon)
 (0.7.3)
 Requirement already satisfied: ujson<6,>=5 in /opt/conda/lib/python3.10/site-
 packages (from
 autogluon.timeseries==0.8.2->autogluon.timeseries[all]==0.8.2->autogluon)
 (5.8.0)
 Requirement already satisfied: psutil<6,>=5.7.3 in
 /opt/conda/lib/python3.10/site-packages (from autogluon.common==0.8.2-
 >autogluon.core==0.8.2->autogluon.core[all]==0.8.2->autogluon) (5.9.3)
 Requirement already satisfied: setuptools in /opt/conda/lib/python3.10/site-
 packages (from autogluon.common==0.8.2->autogluon.core==0.8.2-
 >autogluon.core[all]==0.8.2->autogluon) (60.2.0)
 Requirement already satisfied: contourpy>=1.0.1 in
 /opt/conda/lib/python3.10/site-packages (from matplotlib) (1.1.0)
 Requirement already satisfied: cycycler>=0.10 in /opt/conda/lib/python3.10/site-
 packages (from matplotlib) (0.11.0)
 Requirement already satisfied: fonttools>=4.22.0 in
 /opt/conda/lib/python3.10/site-packages (from matplotlib) (4.42.1)
 Requirement already satisfied: kiwisolver>=1.0.1 in
 /opt/conda/lib/python3.10/site-packages (from matplotlib) (1.4.4)
 Requirement already satisfied: packaging>=20.0 in
 /opt/conda/lib/python3.10/site-packages (from matplotlib) (23.1)
 Requirement already satisfied: pyparsing>=2.2.1 in

/opt/conda/lib/python3.10/site-packages (from matplotlib) (3.0.9)
 Requirement already satisfied: python-dateutil>=2.7 in
 /opt/conda/lib/python3.10/site-packages (from matplotlib) (2.8.2)
 Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.10/site-
 packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
 Requirement already satisfied: pyyaml in /opt/conda/lib/python3.10/site-packages
 (from accelerate<0.17,>=0.9->autogluon.multimodal==0.8.2->autogluon) (6.0.1)
 Requirement already satisfied: botocore<1.32.0,>=1.31.68 in
 /opt/conda/lib/python3.10/site-packages (from
 boto3<2,>=1.10->autogluon.core==0.8.2->autogluon.core[all]==0.8.2->autogluon)
 (1.31.68)
 Requirement already satisfied: jmespath<2.0.0,>=0.7.1 in
 /opt/conda/lib/python3.10/site-packages (from
 boto3<2,>=1.10->autogluon.core==0.8.2->autogluon.core[all]==0.8.2->autogluon)
 (0.10.0)
 Requirement already satisfied: s3transfer<0.8.0,>=0.7.0 in
 /opt/conda/lib/python3.10/site-packages (from
 boto3<2,>=1.10->autogluon.core==0.8.2->autogluon.core[all]==0.8.2->autogluon)
 (0.7.0)
 Requirement already satisfied: graphviz in /opt/conda/lib/python3.10/site-
 packages (from catboost<1.3,>=1.1->autogluon.tabular[all]==0.8.2->autogluon)
 (0.20.1)
 Requirement already satisfied: plotly in /opt/conda/lib/python3.10/site-packages
 (from catboost<1.3,>=1.1->autogluon.tabular[all]==0.8.2->autogluon) (5.16.1)
 Requirement already satisfied: datasets>=2.0.0 in
 /opt/conda/lib/python3.10/site-packages (from
 evaluate<0.4.0,>=0.2.2->autogluon.multimodal==0.8.2->autogluon) (2.14.5)
 Requirement already satisfied: dill in /opt/conda/lib/python3.10/site-packages
 (from evaluate<0.4.0,>=0.2.2->autogluon.multimodal==0.8.2->autogluon) (0.3.7)
 Requirement already satisfied: xxhash in /opt/conda/lib/python3.10/site-packages
 (from evaluate<0.4.0,>=0.2.2->autogluon.multimodal==0.8.2->autogluon) (3.4.1)
 Requirement already satisfied: multiprocessing in /opt/conda/lib/python3.10/site-
 packages (from evaluate<0.4.0,>=0.2.2->autogluon.multimodal==0.8.2->autogluon)
 (0.70.15)
 Requirement already satisfied: fsspec>=2021.05.0 in
 /opt/conda/lib/python3.10/site-packages (from fsspec[http]>=2021.05.0-
 >evaluate<0.4.0,>=0.2.2->autogluon.multimodal==0.8.2->autogluon) (2023.6.0)
 Requirement already satisfied: huggingface-hub>=0.7.0 in
 /opt/conda/lib/python3.10/site-packages (from
 evaluate<0.4.0,>=0.2.2->autogluon.multimodal==0.8.2->autogluon) (0.18.0)
 Requirement already satisfied: responses<0.19 in /opt/conda/lib/python3.10/site-
 packages (from evaluate<0.4.0,>=0.2.2->autogluon.multimodal==0.8.2->autogluon)
 (0.18.0)
 Requirement already satisfied: pip in /opt/conda/lib/python3.10/site-packages
 (from fastai<2.8,>=2.3.1->autogluon.tabular[all]==0.8.2->autogluon) (23.3.1)
 Requirement already satisfied: fastdownload<2,>=0.0.5 in
 /opt/conda/lib/python3.10/site-packages (from
 fastai<2.8,>=2.3.1->autogluon.tabular[all]==0.8.2->autogluon) (0.0.7)

Requirement already satisfied: fastcore<1.6,>=1.5.29 in /opt/conda/lib/python3.10/site-packages (from fastai<2.8,>=2.3.1->autogluon.tabular[all]==0.8.2->autogluon) (1.5.29)

Requirement already satisfied: fastprogress>=0.2.4 in /opt/conda/lib/python3.10/site-packages (from fastai<2.8,>=2.3.1->autogluon.tabular[all]==0.8.2->autogluon) (1.0.3)

Requirement already satisfied: spacy<4 in /opt/conda/lib/python3.10/site-packages (from fastai<2.8,>=2.3.1->autogluon.tabular[all]==0.8.2->autogluon) (3.7.2)

Requirement already satisfied: toolz~=0.10 in /opt/conda/lib/python3.10/site-packages (from gluonts<0.14,>=0.13.1->autogluon.timeseries==0.8.2->autogluon.timeseries[all]==0.8.2->autogluon) (0.12.0)

Requirement already satisfied: typing-extensions~=4.0 in /opt/conda/lib/python3.10/site-packages (from gluonts<0.14,>=0.13.1->autogluon.timeseries==0.8.2->autogluon.timeseries[all]==0.8.2->autogluon) (4.7.1)

Requirement already satisfied: future in /opt/conda/lib/python3.10/site-packages (from hyperopt<0.2.8,>=0.2.7->autogluon.core[all]==0.8.2->autogluon) (0.18.3)

Requirement already satisfied: cloudpickle in /opt/conda/lib/python3.10/site-packages (from hyperopt<0.2.8,>=0.2.7->autogluon.core[all]==0.8.2->autogluon) (2.2.1)

Requirement already satisfied: py4j in /opt/conda/lib/python3.10/site-packages (from hyperopt<0.2.8,>=0.2.7->autogluon.core[all]==0.8.2->autogluon) (0.10.9.7)

Requirement already satisfied: MarkupSafe>=2.0 in /opt/conda/lib/python3.10/site-packages (from jinja2<3.2,>=3.0.3->autogluon.multimodal==0.8.2->autogluon) (2.1.3)

Requirement already satisfied: attrs>=17.4.0 in /opt/conda/lib/python3.10/site-packages (from jsonschema<4.18,>=4.14->autogluon.multimodal==0.8.2->autogluon) (23.1.0)

Requirement already satisfied: pyrsistent!=0.17.0,!0.17.1,!0.17.2,>=0.14.0 in /opt/conda/lib/python3.10/site-packages (from jsonschema<4.18,>=4.14->autogluon.multimodal==0.8.2->autogluon) (0.19.3)

Requirement already satisfied: wheel in /opt/conda/lib/python3.10/site-packages (from lightgbm<3.4,>=3.3->autogluon.tabular[all]==0.8.2->autogluon) (0.41.2)

Requirement already satisfied: numba in /opt/conda/lib/python3.10/site-packages (from mlforecast<0.7.4,>=0.7.0->autogluon.timeseries==0.8.2->autogluon.timeseries[all]==0.8.2->autogluon) (0.57.1)

Requirement already satisfied: window-ops in /opt/conda/lib/python3.10/site-packages (from mlforecast<0.7.4,>=0.7.0->autogluon.timeseries==0.8.2->autogluon.timeseries[all]==0.8.2->autogluon) (0.0.14)

Requirement already satisfied: gdown>=4.0.0 in /opt/conda/lib/python3.10/site-packages (from nlpaug<1.2.0,>=1.1.10->autogluon.multimodal==0.8.2->autogluon) (4.7.1)

Requirement already satisfied: click in /opt/conda/lib/python3.10/site-packages (from nltk<4.0.0,>=3.4.5->autogluon.multimodal==0.8.2->autogluon) (8.1.7)

Requirement already satisfied: regex>=2021.8.3 in /opt/conda/lib/python3.10/site-packages (from nltk<4.0.0,>=3.4.5->autogluon.multimodal==0.8.2->autogluon) (2023.10.3)

Requirement already satisfied: antlr4-python3-runtime==4.9.* in /opt/conda/lib/python3.10/site-packages (from omegaconf<2.3.0,>=2.1.1->autogluon.multimodal==0.8.2->autogluon) (4.9.3)

Requirement already satisfied: colorama in /opt/conda/lib/python3.10/site-packages (from openmim<0.4.0,>=0.3.7->autogluon.multimodal==0.8.2->autogluon) (0.4.6)

Requirement already satisfied: model-index in /opt/conda/lib/python3.10/site-packages (from openmim<0.4.0,>=0.3.7->autogluon.multimodal==0.8.2->autogluon) (0.1.11)

Requirement already satisfied: opendatalab in /opt/conda/lib/python3.10/site-packages (from openmim<0.4.0,>=0.3.7->autogluon.multimodal==0.8.2->autogluon) (0.0.10)

Requirement already satisfied: rich in /opt/conda/lib/python3.10/site-packages (from openmim<0.4.0,>=0.3.7->autogluon.multimodal==0.8.2->autogluon) (13.4.2)

Requirement already satisfied: tabulate in /opt/conda/lib/python3.10/site-packages (from openmim<0.4.0,>=0.3.7->autogluon.multimodal==0.8.2->autogluon) (0.9.0)

Requirement already satisfied: pytz>=2020.1 in /opt/conda/lib/python3.10/site-packages (from pandas<1.6,>=1.4.1->autogluon.core==0.8.2->autogluon.core[all]==0.8.2->autogluon) (2023.3)

Requirement already satisfied: lightning-utilities>=0.6.0.post0 in /opt/conda/lib/python3.10/site-packages (from pytorch-lightning<1.10.0,>=1.9.0->autogluon.multimodal==0.8.2->autogluon) (0.9.0)

Requirement already satisfied: filelock in /opt/conda/lib/python3.10/site-packages (from ray<2.4,>=2.3->ray[default]<2.4,>=2.3; extra == "all"->autogluon.core[all]==0.8.2->autogluon) (3.12.2)

Requirement already satisfied: msgpack<2.0.0,>=1.0.0 in /opt/conda/lib/python3.10/site-packages (from ray<2.4,>=2.3->ray[default]<2.4,>=2.3; extra == "all"->autogluon.core[all]==0.8.2->autogluon) (1.0.5)

Requirement already satisfied: protobuf!=3.19.5,>=3.15.3 in /opt/conda/lib/python3.10/site-packages (from ray<2.4,>=2.3->ray[default]<2.4,>=2.3; extra == "all"->autogluon.core[all]==0.8.2->autogluon) (3.20.2)

Requirement already satisfied: aiosignal in /opt/conda/lib/python3.10/site-packages (from ray<2.4,>=2.3->ray[default]<2.4,>=2.3; extra == "all"->autogluon.core[all]==0.8.2->autogluon) (1.3.1)

Requirement already satisfied: frozenlist in /opt/conda/lib/python3.10/site-packages (from ray<2.4,>=2.3->ray[default]<2.4,>=2.3; extra == "all"->autogluon.core[all]==0.8.2->autogluon) (1.4.0)

Requirement already satisfied: virtualenv>=20.0.24 in /opt/conda/lib/python3.10/site-packages (from ray<2.4,>=2.3->ray[default]<2.4,>=2.3; extra == "all"->autogluon.core[all]==0.8.2->autogluon) (20.21.0)

Requirement already satisfied: aiohttp>=3.7 in /opt/conda/lib/python3.10/site-packages (from ray[default]<2.4,>=2.3; extra == "all"->autogluon.core[all]==0.8.2->autogluon) (3.8.5)

Requirement already satisfied: aiohttp-cors in /opt/conda/lib/python3.10/site-

```

packages (from ray[default]<2.4,>=2.3; extra ==
"all"->autogluon.core[all]==0.8.2->autogluon) (0.7.0)
Requirement already satisfied: colorful in /opt/conda/lib/python3.10/site-
packages (from ray[default]<2.4,>=2.3; extra ==
"all"->autogluon.core[all]==0.8.2->autogluon) (0.5.5)
Requirement already satisfied: py-spy>=0.2.0 in /opt/conda/lib/python3.10/site-
packages (from ray[default]<2.4,>=2.3; extra ==
"all"->autogluon.core[all]==0.8.2->autogluon) (0.3.14)
Requirement already satisfied: gpustat>=1.0.0 in /opt/conda/lib/python3.10/site-
packages (from ray[default]<2.4,>=2.3; extra ==
"all"->autogluon.core[all]==0.8.2->autogluon) (1.0.0)
Requirement already satisfied: opencensus in /opt/conda/lib/python3.10/site-
packages (from ray[default]<2.4,>=2.3; extra ==
"all"->autogluon.core[all]==0.8.2->autogluon) (0.11.2)
Requirement already satisfied: prometheus-client>=0.7.1 in
/opt/conda/lib/python3.10/site-packages (from ray[default]<2.4,>=2.3; extra ==
"all"->autogluon.core[all]==0.8.2->autogluon) (0.17.1)
Requirement already satisfied: smart-open in /opt/conda/lib/python3.10/site-
packages (from ray[default]<2.4,>=2.3; extra ==
"all"->autogluon.core[all]==0.8.2->autogluon) (6.3.0)
Requirement already satisfied: tensorboardX>=1.9 in
/opt/conda/lib/python3.10/site-packages (from ray[tune]<2.4,>=2.3; extra ==
"all"->autogluon.core[all]==0.8.2->autogluon) (2.6.2.2)
Requirement already satisfied: charset-normalizer<4,>=2 in
/opt/conda/lib/python3.10/site-packages (from
requests->autogluon.core==0.8.2->autogluon.core[all]==0.8.2->autogluon) (3.2.0)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/lib/python3.10/site-
packages (from
requests->autogluon.core==0.8.2->autogluon.core[all]==0.8.2->autogluon) (3.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
/opt/conda/lib/python3.10/site-packages (from
requests->autogluon.core==0.8.2->autogluon.core[all]==0.8.2->autogluon)
(1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in
/opt/conda/lib/python3.10/site-packages (from
requests->autogluon.core==0.8.2->autogluon.core[all]==0.8.2->autogluon)
(2023.7.22)
Requirement already satisfied: imageio>=2.4.1 in /opt/conda/lib/python3.10/site-
packages (from scikit-
image<0.20.0,>=0.19.1->autogluon.multimodal==0.8.2->autogluon) (2.31.1)
Requirement already satisfied: tifffile>=2019.7.26 in
/opt/conda/lib/python3.10/site-packages (from scikit-
image<0.20.0,>=0.19.1->autogluon.multimodal==0.8.2->autogluon) (2023.8.12)
Requirement already satisfied: PyWavelets>=1.1.1 in
/opt/conda/lib/python3.10/site-packages (from scikit-
image<0.20.0,>=0.19.1->autogluon.multimodal==0.8.2->autogluon) (1.4.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in
/opt/conda/lib/python3.10/site-packages (from scikit-

```



```

learn<1.3,>=1.0->autogluon.core==0.8.2->autogluon.core[all]==0.8.2->autogluon)
(3.2.0)
Requirement already satisfied: patsy>=0.5.2 in /opt/conda/lib/python3.10/site-
packages (from statsmodels<0.15,>=0.13.0->autogluon.timeseries==0.8.2-
>autogluon.timeseries[all]==0.8.2->autogluon) (0.5.3)
Requirement already satisfied: absl-py>=0.4 in /opt/conda/lib/python3.10/site-
packages (from tensorboard<3,>=2.9->autogluon.multimodal==0.8.2->autogluon)
(1.4.0)
Requirement already satisfied: google-auth<3,>=1.6.3 in
/opt/conda/lib/python3.10/site-packages (from
tensorboard<3,>=2.9->autogluon.multimodal==0.8.2->autogluon) (2.22.0)
Requirement already satisfied: google-auth-oauthlib<2,>=0.5 in
/opt/conda/lib/python3.10/site-packages (from
tensorboard<3,>=2.9->autogluon.multimodal==0.8.2->autogluon) (1.0.0)
Requirement already satisfied: markdown>=2.6.8 in
/opt/conda/lib/python3.10/site-packages (from
tensorboard<3,>=2.9->autogluon.multimodal==0.8.2->autogluon) (3.5)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in
/opt/conda/lib/python3.10/site-packages (from
tensorboard<3,>=2.9->autogluon.multimodal==0.8.2->autogluon) (0.7.1)
Requirement already satisfied: werkzeug>=1.0.1 in
/opt/conda/lib/python3.10/site-packages (from
tensorboard<3,>=2.9->autogluon.multimodal==0.8.2->autogluon) (3.0.0)
Requirement already satisfied: safetensors in /opt/conda/lib/python3.10/site-
packages (from timm<0.10.0,>=0.9.2->autogluon.multimodal==0.8.2->autogluon)
(0.4.0)
Requirement already satisfied: tokenizers!=0.11.3,<0.14,>=0.11.1 in
/opt/conda/lib/python3.10/site-packages (from transformers<4.27.0,>=4.23.0-
>transformers[sentencepiece]<4.27.0,>=4.23.0->autogluon.multimodal==0.8.2-
>autogluon) (0.13.3)
Requirement already satisfied: sentencepiece!=0.1.92,>=0.1.91 in
/opt/conda/lib/python3.10/site-packages (from transformers[sentencepiece]<4.27.0
,>=4.23.0->autogluon.multimodal==0.8.2->autogluon) (0.1.99)
Requirement already satisfied: multidict<7.0,>=4.5 in
/opt/conda/lib/python3.10/site-packages (from
aiohttp>=3.7->ray[default]<2.4,>=2.3; extra ==
"all"->autogluon.core[all]==0.8.2->autogluon) (6.0.4)
Requirement already satisfied: async-timeout<5.0,>=4.0.0a3 in
/opt/conda/lib/python3.10/site-packages (from
aiohttp>=3.7->ray[default]<2.4,>=2.3; extra ==
"all"->autogluon.core[all]==0.8.2->autogluon) (4.0.3)
Requirement already satisfied: yarl<2.0,>=1.0 in /opt/conda/lib/python3.10/site-
packages (from aiohttp>=3.7->ray[default]<2.4,>=2.3; extra ==
"all"->autogluon.core[all]==0.8.2->autogluon) (1.9.2)
Requirement already satisfied: pyarrow>=8.0.0 in /opt/conda/lib/python3.10/site-
packages (from
datasets>=2.0.0->evaluate<0.4.0,>=0.2.2->autogluon.multimodal==0.8.2->autogluon)
(12.0.1)

```

Requirement already satisfied: beautifulsoup4 in /opt/conda/lib/python3.10/site-packages (from gdown>=4.0.0->nlpaug<1.2.0,>=1.1.10->autogluon.multimodal==0.8.2->autogluon) (4.12.2)

Requirement already satisfied: cachetools<6.0,>=2.0.0 in /opt/conda/lib/python3.10/site-packages (from google-auth<3,>=1.6.3->tensorboard<3,>=2.9->autogluon.multimodal==0.8.2->autogluon) (5.3.1)

Requirement already satisfied: pyasn1-modules>=0.2.1 in /opt/conda/lib/python3.10/site-packages (from google-auth<3,>=1.6.3->tensorboard<3,>=2.9->autogluon.multimodal==0.8.2->autogluon) (0.3.0)

Requirement already satisfied: rsa<5,>=3.1.4 in /opt/conda/lib/python3.10/site-packages (from google-auth<3,>=1.6.3->tensorboard<3,>=2.9->autogluon.multimodal==0.8.2->autogluon) (4.9)

Requirement already satisfied: requests-oauthlib>=0.7.0 in /opt/conda/lib/python3.10/site-packages (from google-auth-oauthlib<2,>=0.5->tensorboard<3,>=2.9->autogluon.multimodal==0.8.2->autogluon) (1.3.1)

Requirement already satisfied: nvidia-ml-py<=11.495.46,>=11.450.129 in /opt/conda/lib/python3.10/site-packages (from gpustat>=1.0.0->ray[default]<2.4,>=2.3; extra == "all"->autogluon.core[all]==0.8.2->autogluon) (11.495.46)

Requirement already satisfied: blessed>=1.17.1 in /opt/conda/lib/python3.10/site-packages (from gpustat>=1.0.0->ray[default]<2.4,>=2.3; extra == "all"->autogluon.core[all]==0.8.2->autogluon) (1.20.0)

Requirement already satisfied: llvmlite<0.41,>=0.40.0dev0 in /opt/conda/lib/python3.10/site-packages (from numba->mlforecast<0.7.4,>=0.7.0->autogluon.timeseries==0.8.2->autogluon.timeseries[all]==0.8.2->autogluon) (0.40.1)

Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in /opt/conda/lib/python3.10/site-packages (from spacy<4->fastai<2.8,>=2.3.1->autogluon.tabular[all]==0.8.2->autogluon) (3.0.12)

Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /opt/conda/lib/python3.10/site-packages (from spacy<4->fastai<2.8,>=2.3.1->autogluon.tabular[all]==0.8.2->autogluon) (1.0.5)

Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /opt/conda/lib/python3.10/site-packages (from spacy<4->fastai<2.8,>=2.3.1->autogluon.tabular[all]==0.8.2->autogluon) (1.0.10)

Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /opt/conda/lib/python3.10/site-packages (from spacy<4->fastai<2.8,>=2.3.1->autogluon.tabular[all]==0.8.2->autogluon) (2.0.8)

Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /opt/conda/lib/python3.10/site-packages (from spacy<4->fastai<2.8,>=2.3.1->autogluon.tabular[all]==0.8.2->autogluon) (3.0.9)

Requirement already satisfied: thinc<8.3.0,>=8.1.8 in

```

/opt/conda/lib/python3.10/site-packages (from
spacy<4->fastai<2.8,>=2.3.1->autogluon.tabular[all]==0.8.2->autogluon) (8.2.1)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in
/opt/conda/lib/python3.10/site-packages (from
spacy<4->fastai<2.8,>=2.3.1->autogluon.tabular[all]==0.8.2->autogluon) (1.1.2)
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in
/opt/conda/lib/python3.10/site-packages (from
spacy<4->fastai<2.8,>=2.3.1->autogluon.tabular[all]==0.8.2->autogluon) (2.4.8)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in
/opt/conda/lib/python3.10/site-packages (from
spacy<4->fastai<2.8,>=2.3.1->autogluon.tabular[all]==0.8.2->autogluon) (2.0.10)
Requirement already satisfied: weasel<0.4.0,>=0.1.0 in
/opt/conda/lib/python3.10/site-packages (from
spacy<4->fastai<2.8,>=2.3.1->autogluon.tabular[all]==0.8.2->autogluon) (0.3.3)
Requirement already satisfied: typer<0.10.0,>=0.3.0 in
/opt/conda/lib/python3.10/site-packages (from
spacy<4->fastai<2.8,>=2.3.1->autogluon.tabular[all]==0.8.2->autogluon) (0.9.0)
Requirement already satisfied: langcodes<4.0.0,>=3.2.0 in
/opt/conda/lib/python3.10/site-packages (from
spacy<4->fastai<2.8,>=2.3.1->autogluon.tabular[all]==0.8.2->autogluon) (3.3.0)
Requirement already satisfied: distlib<1,>=0.3.6 in
/opt/conda/lib/python3.10/site-packages (from
virtualenv>=20.0.24->ray<2.4,>=2.3->ray[default]<2.4,>=2.3; extra ==
"all"->autogluon.core[all]==0.8.2->autogluon) (0.3.7)
Requirement already satisfied: platformdirs<4,>=2.4 in
/opt/conda/lib/python3.10/site-packages (from
virtualenv>=20.0.24->ray<2.4,>=2.3->ray[default]<2.4,>=2.3; extra ==
"all"->autogluon.core[all]==0.8.2->autogluon) (3.10.0)
Requirement already satisfied: ordered-set in /opt/conda/lib/python3.10/site-
packages (from model-
index->openmim<0.4.0,>=0.3.7->autogluon.multimodal==0.8.2->autogluon) (4.1.0)
Requirement already satisfied: opencensus-context>=0.1.3 in
/opt/conda/lib/python3.10/site-packages (from
opencensus->ray[default]<2.4,>=2.3; extra ==
"all"->autogluon.core[all]==0.8.2->autogluon) (0.1.3)
Requirement already satisfied: google-api-core<3.0.0,>=1.0.0 in
/opt/conda/lib/python3.10/site-packages (from
opencensus->ray[default]<2.4,>=2.3; extra ==
"all"->autogluon.core[all]==0.8.2->autogluon) (1.34.0)
Requirement already satisfied: pycryptodome in /opt/conda/lib/python3.10/site-
packages (from
opendatalab->openmim<0.4.0,>=0.3.7->autogluon.multimodal==0.8.2->autogluon)
(3.19.0)
Requirement already satisfied: openxlab in /opt/conda/lib/python3.10/site-
packages (from
opendatalab->openmim<0.4.0,>=0.3.7->autogluon.multimodal==0.8.2->autogluon)
(0.0.28)
Requirement already satisfied: tenacity>=6.2.0 in

```

/opt/conda/lib/python3.10/site-packages (from
 plotly->catboost<1.3,>=1.1->autogluon.tabular[all]==0.8.2->autogluon) (8.2.3)
 Requirement already satisfied: markdown-it-py>=2.2.0 in
 /opt/conda/lib/python3.10/site-packages (from
 rich->openmim<0.4.0,>=0.3.7->autogluon.multimodal==0.8.2->autogluon) (3.0.0)
 Requirement already satisfied: pygments<3.0.0,>=2.13.0 in
 /opt/conda/lib/python3.10/site-packages (from
 rich->openmim<0.4.0,>=0.3.7->autogluon.multimodal==0.8.2->autogluon) (2.16.1)
 Requirement already satisfied: wcwidth>=0.1.4 in /opt/conda/lib/python3.10/site-
 packages (from blessed>=1.17.1->gpustat>=1.0.0->ray[default]<2.4,>=2.3; extra ==
 "all"->autogluon.core[all]==0.8.2->autogluon) (0.2.6)
 Requirement already satisfied: googleapis-common-protos<2.0dev,>=1.56.2 in
 /opt/conda/lib/python3.10/site-packages (from google-api-
 core<3.0.0,>=1.0.0->opencensus->ray[default]<2.4,>=2.3; extra ==
 "all"->autogluon.core[all]==0.8.2->autogluon) (1.60.0)
 Requirement already satisfied: mdurl~=0.1 in /opt/conda/lib/python3.10/site-
 packages (from markdown-it-
 py>=2.2.0->rich->openmim<0.4.0,>=0.3.7->autogluon.multimodal==0.8.2->autogluon)
 (0.1.2)
 Requirement already satisfied: pyasn1<0.6.0,>=0.4.6 in
 /opt/conda/lib/python3.10/site-packages (from pyasn1-modules>=0.2.1->google-
 auth<3,>=1.6.3->tensorboard<3,>=2.9->autogluon.multimodal==0.8.2->autogluon)
 (0.5.0)
 Requirement already satisfied: oauthlib>=3.0.0 in
 /opt/conda/lib/python3.10/site-packages (from requests-oauthlib>=0.7.0->google-
 auth-
 oauthlib<2,>=0.5->tensorboard<3,>=2.9->autogluon.multimodal==0.8.2->autogluon)
 (3.2.2)
 Requirement already satisfied: blis<0.8.0,>=0.7.8 in
 /opt/conda/lib/python3.10/site-packages (from thinc<8.3.0,>=8.1.8->spacy<4-
 >fastai<2.8,>=2.3.1->autogluon.tabular[all]==0.8.2->autogluon) (0.7.11)
 Requirement already satisfied: confection<1.0.0,>=0.0.1 in
 /opt/conda/lib/python3.10/site-packages (from thinc<8.3.0,>=8.1.8->spacy<4-
 >fastai<2.8,>=2.3.1->autogluon.tabular[all]==0.8.2->autogluon) (0.1.3)
 Requirement already satisfied: cloudpathlib<0.17.0,>=0.7.0 in
 /opt/conda/lib/python3.10/site-packages (from weasel<0.4.0,>=0.1.0->spacy<4-
 >fastai<2.8,>=2.3.1->autogluon.tabular[all]==0.8.2->autogluon) (0.16.0)
 Requirement already satisfied: soupsieve>1.2 in /opt/conda/lib/python3.10/site-
 packages (from beautifulsoup4->gdown>=4.0.0->nlpaug<1.2.0,>=1.1.10-
 >autogluon.multimodal==0.8.2->autogluon) (2.3.2.post1)
 Requirement already satisfied: oss2~=2.17.0 in /opt/conda/lib/python3.10/site-
 packages (from openxlab->opendatalab->openmim<0.4.0,>=0.3.7-
 >autogluon.multimodal==0.8.2->autogluon) (2.17.0)
 Requirement already satisfied: PySocks!=1.5.7,>=1.5.6 in
 /opt/conda/lib/python3.10/site-packages (from requests[socks]->gdown>=4.0.0-
 >nlpaug<1.2.0,>=1.1.10->autogluon.multimodal==0.8.2->autogluon) (1.7.1)
 Requirement already satisfied: crcmod>=1.7 in /opt/conda/lib/python3.10/site-
 packages (from oss2~=2.17.0->openxlab->opendatalab->openmim<0.4.0,>=0.3.7-

```

>autogluon.multimodal==0.8.2->autogluon) (1.7)
Requirement already satisfied: aliyun-python-sdk-kms>=2.4.1 in
/opt/conda/lib/python3.10/site-packages (from oss2~=2.17.0->openxlab-
>opendatalab->openmim<0.4.0,>=0.3.7->autogluon.multimodal==0.8.2->autogluon)
(2.16.2)
Requirement already satisfied: aliyun-python-sdk-core>=2.13.12 in
/opt/conda/lib/python3.10/site-packages (from oss2~=2.17.0->openxlab-
>opendatalab->openmim<0.4.0,>=0.3.7->autogluon.multimodal==0.8.2->autogluon)
(2.14.0)
Requirement already satisfied: cryptography>=2.6.0 in
/opt/conda/lib/python3.10/site-packages (from aliyun-python-sdk-core>=2.13.12-
>oss2~=2.17.0->openxlab->opendatalab->openmim<0.4.0,>=0.3.7-
>autogluon.multimodal==0.8.2->autogluon) (41.0.3)
Requirement already satisfied: cffi>=1.12 in /opt/conda/lib/python3.10/site-
packages (from cryptography>=2.6.0->aliyun-python-sdk-core>=2.13.12-
>oss2~=2.17.0->openxlab->opendatalab->openmim<0.4.0,>=0.3.7-
>autogluon.multimodal==0.8.2->autogluon) (1.15.1)
Requirement already satisfied: pycparser in /opt/conda/lib/python3.10/site-
packages (from cffi>=1.12->cryptography>=2.6.0->aliyun-python-sdk-core>=2.13.12-
>oss2~=2.17.0->openxlab->opendatalab->openmim<0.4.0,>=0.3.7-
>autogluon.multimodal==0.8.2->autogluon) (2.21)

```

2 Config

```

[1]: # config

label = 'y'
metric = 'mean_absolute_error'
time_limit = None
presets = "experimental_zeroshot_hpo_hybrid"#'best_quality'

use_is_estimated_attr = True

drop_night_outliers = True

# to_drop = ["snow_drift:idx", "snow_density:kgm3", "wind_speed_w_1000hPa:ms",
↳ "dew_or_rime:idx", "prob_rime:p", "fresh_snow_12h:cm", "fresh_snow_24h:cm",
↳ "wind_speed_u_10m:ms", "wind_speed_v_10m:ms", "snow_melt_10min:mm",
↳ "rain_water:kgm2", "dew_point_2m:K", "precip_5min:mm", "absolute_humidity_2m:
↳ gm3", "air_density_2m:kgm3"]#, "msl_pressure:hPa", "pressure_50m:hPa",
↳ "pressure_100m:hPa"]
to_drop = ["wind_speed_w_1000hPa:ms", "wind_speed_u_10m:ms", "wind_speed_v_10m:
↳ ms"]

excluded_model_types = ['CAT', 'XGB', 'RF']

```

```

num_stack_levels = 0
num_bag_folds = 8##8# 8
num_bag_sets = 3##20

use_tune_data = True
use_test_data = True
use_bag_holdout = True

clip_predictions = True

```

3 Loading and preprocessing

```

[2]: import pandas as pd
import numpy as np

import warnings
warnings.filterwarnings("ignore")

def feature_engineering(X):
    # shift columns with them by 1 hour, so that for index 16:00, we have the
    ↪ values from 17:00
    columns = ['clear_sky_energy_1h:J', 'diffuse_rad_1h:J', 'direct_rad_1h:J',
               'fresh_snow_12h:cm', 'fresh_snow_1h:cm', 'fresh_snow_24h:cm',
               'fresh_snow_3h:cm', 'fresh_snow_6h:cm']

    # Filter rows where index.minute == 0
    X_shifted = X[X.index.minute == 0][columns].copy()

    # Create a set for constant-time lookup
    index_set = set(X.index)

    # Vectorized time shifting
    one_hour = pd.Timedelta('1 hour')
    shifted_indices = X_shifted.index + one_hour
    X_shifted.loc[shifted_indices.isin(index_set)] = X.
    ↪ loc[shifted_indices[shifted_indices.isin(index_set)]] [columns]

    # set last row to same as second last row
    X_shifted.iloc[-1] = X_shifted.iloc[-2]

    # Rename columns
    X_old_unshifted = X_shifted.copy()

```

```

X_old_unshifted.columns = [f"{col}_not_shifted" for col in X_old_unshifted.
↪columns]

date_calc = None
# If 'date_calc' is present, handle it
if 'date_calc' in X.columns:
    date_calc = X[X.index.minute == 0]['date_calc']

# resample to hourly
X = X.resample('H').mean()

# overwrite columns with shifted columns
X[columns] = X_shifted[columns]

if date_calc is not None:
    X['date_calc'] = date_calc

return X

def fix_X(X, name):
    # Convert 'date_forecast' to datetime format and replace original column
    ↪with 'ds'
    X['ds'] = pd.to_datetime(X['date_forecast'])
    X.drop(columns=['date_forecast'], inplace=True, errors='ignore')
    X.sort_values(by='ds', inplace=True)
    X.set_index('ds', inplace=True)

    X = feature_engineering(X)

    return X

def handle_features(X_train_observed, X_train_estimated, X_test, y_train):
    X_train_observed = fix_X(X_train_observed, "X_train_observed")
    X_train_estimated = fix_X(X_train_estimated, "X_train_estimated")
    X_test = fix_X(X_test, "X_test")

    y_train['ds'] = pd.to_datetime(y_train['time'])

```

```

y_train.drop(columns=['time'], inplace=True)
y_train.sort_values(by='ds', inplace=True)
y_train.set_index('ds', inplace=True)

return X_train_observed, X_train_estimated, X_test, y_train

def preprocess_data(X_train_observed, X_train_estimated, X_test, y_train,
location):
    # convert to datetime
    X_train_observed, X_train_estimated, X_test, y_train =
handle_features(X_train_observed, X_train_estimated, X_test, y_train)

    if use_is_estimated_attr:
        X_train_observed["is_estimated"] = 0
        X_train_estimated["is_estimated"] = 1
        X_test["is_estimated"] = 1

    # drop date_calc
    X_train_estimated.drop(columns=['date_calc'], inplace=True)
    X_test.drop(columns=['date_calc'], inplace=True)

    y_train["y"] = y_train["pv_measurement"].astype('float64')
    y_train.drop(columns=['pv_measurement'], inplace=True)
    X_train = pd.concat([X_train_observed, X_train_estimated])

    # clip all y values to 0 if negative
    y_train["y"] = y_train["y"].clip(lower=0)

    X_train = pd.merge(X_train, y_train, how="inner", left_index=True,
right_index=True)

    X_train["location"] = location
    X_test["location"] = location

    return X_train, X_test

# Define locations
locations = ['A', 'B', 'C']

X_trains = []
X_tests = []
# Loop through locations

```



```

for loc in locations:
    print(f"Processing location {loc}...")
    # Read target training data
    y_train = pd.read_parquet(f'{loc}/train_targets.parquet')

    # Read estimated training data and add location feature
    X_train_estimated = pd.read_parquet(f'{loc}/X_train_estimated.parquet')

    # Read observed training data and add location feature
    X_train_observed = pd.read_parquet(f'{loc}/X_train_observed.parquet')

    # Read estimated test data and add location feature
    X_test_estimated = pd.read_parquet(f'{loc}/X_test_estimated.parquet')

    # Preprocess data
    X_train, X_test = preprocess_data(X_train_observed, X_train_estimated,
    ↪X_test_estimated, y_train, loc)

    X_trains.append(X_train)
    X_tests.append(X_test)

# Concatenate all data and save to csv
X_train = pd.concat(X_trains)
X_test = pd.concat(X_tests)

```

Processing location A...

Processing location B...

Processing location C...

3.1 Feature engineering

3.1.1 Remove anomalies

```

[3]: import pandas as pd

def replace_streaks_with_nan(df, max_streak_length, column="y"):
    for location in df["location"].unique():
        x = df[df["location"] == location][column].copy()

        last_val = None
        streak_length = 1
        streak_indices = []
        allowed = [0]
        found_streaks = {}

        for idx in x.index:
            value = x[idx]

```

```

    if value == last_val and value not in allowed:
        streak_length += 1
        streak_indices.append(idx)
    else:
        streak_length = 1
        last_val = value
        streak_indices.clear()

    if streak_length > max_streak_length:
        found_streaks[value] = streak_length

        for streak_idx in streak_indices:
            x[idx] = np.nan
            streak_indices.clear() # clear after setting to NaN to avoid
↪setting multiple times
        df.loc[df["location"] == location, column] = x

    print(f"Found streaks for location {location}: {found_streaks}")

    return df

```

```
X_train = replace_streaks_with_nan(X_train.copy(), 3, "y")
```

Found streaks for location A: {}

Found streaks for location B: {3.45: 28, 6.9: 7, 12.9375: 5, 13.8: 8, 276.0: 78, 18.975: 58, 0.8625: 4, 118.1625: 33, 34.5: 11, 183.7125: 1058, 87.1125: 7, 79.35: 34, 7.7625: 12, 27.6: 448, 273.41249999999997: 72, 264.78749999999997: 55, 169.05: 33, 375.1875: 56, 314.8125: 66, 76.7625: 10, 135.4125: 216, 81.9375: 202, 2.5875: 12, 81.075: 210}

Found streaks for location C: {9.8: 4, 29.400000000000002: 4, 19.6: 4}

```

[4]: # print num rows
temprows = len(X_train)
X_train.dropna(subset=['y', 'direct_rad_1h:J', 'diffuse_rad_1h:J'],
↪inplace=True)
print("Dropped rows: ", temprows - len(X_train))

```

Dropped rows: 9293

```

[5]: thresh = 0.1
mask = (X_train["direct_rad_1h:J"] <= thresh) & (X_train["diffuse_rad_1h:J"] <=
↪thresh) & (X_train["y"] >= 0.1)

if drop_night_outliers:
    X_train.loc[mask, "y"] = np.nan

```

```
[6]: temprows = len(X_train)
X_train.dropna(subset=['y', 'direct_rad_1h:J', 'diffuse_rad_1h:J'],
               ↪inplace=True)
print("Dropped rows: ", temprows - len(X_train))
```

Dropped rows: 1876

```
[7]: X_train.drop(columns=to_drop, inplace=True)
X_test.drop(columns=to_drop, inplace=True)

X_train.to_csv('X_train_raw.csv', index=True)
X_test.to_csv('X_test_raw.csv', index=True)
```

```
[8]: def split_and_shuffle_data(input_data, num_bins, frac1):
    # Validate the input fraction
    if frac1 < 0 or frac1 > 1:
        raise ValueError("frac1 must be between 0 and 1.")

    if frac1==1:
        return input_data, pd.DataFrame()

    # Calculate the fraction for the second output set
    frac2 = 1 - frac1

    # Calculate bin size
    bin_size = len(input_data) // num_bins

    # Initialize empty DataFrames for output
    output_data1 = pd.DataFrame()
    output_data2 = pd.DataFrame()

    for i in range(num_bins):
        # Shuffle the data in the current bin
        np.random.seed(i)
        current_bin = input_data.iloc[i * bin_size: (i + 1) * bin_size].
        ↪sample(frac=1)

        # Calculate the sizes for each output set
        size1 = int(len(current_bin) * frac1)

        # Split and append to output DataFrames
        output_data1 = pd.concat([output_data1, current_bin.iloc[:size1]])
        output_data2 = pd.concat([output_data2, current_bin.iloc[size1:]])

    # Shuffle and split the remaining data
    remaining_data = input_data.iloc[num_bins * bin_size:].sample(frac=1)
```

```

    remaining_size1 = int(len(remaining_data) * frac1)

    output_data1 = pd.concat([output_data1, remaining_data.iloc[:
↪remaining_size1]])
    output_data2 = pd.concat([output_data2, remaining_data.iloc[remaining_size1:
↪]])

    return output_data1, output_data2

```

```

[9]: from autogluon.tabular import TabularDataset, TabularPredictor
data = TabularDataset('X_train_raw.csv')
data['ds'] = pd.to_datetime(data['ds'])
data = data.sort_values(by='ds')

split_time = pd.to_datetime("2022-10-28 22:00:00")
train_set = TabularDataset(data[data["ds"] < split_time])
estimated_set = TabularDataset(data[data["ds"] >= split_time]) # only estimated

test_set = pd.DataFrame()
tune_set = pd.DataFrame()
new_train_set = pd.DataFrame()

for location in locations:
    loc_data = data[data["location"] == location]
    num_train_rows = len(loc_data)

    tune_rows = 1500.0 # 2500.0
    if use_test_data:
        tune_rows = 1880.0#max(3000.0,
↪len(estimated_set[estimated_set["location"] == location]))

    holdout_frac = max(0.01, min(0.1, tune_rows / num_train_rows)) *
↪num_train_rows / len(estimated_set[estimated_set["location"] == location])

    # shuffle and split data
    loc_tune_set, loc_new_train_set =
↪split_and_shuffle_data(estimated_set[estimated_set['location'] == location],
↪40, holdout_frac)

    new_train_set = pd.concat([new_train_set, loc_new_train_set])

    if use_test_data:
        loc_test_set, loc_tune_set = split_and_shuffle_data(loc_tune_set, 40, 0.
↪2)

        test_set = pd.concat([test_set, loc_test_set])

    tune_set = pd.concat([tune_set, loc_tune_set])

```

```

# add rest to train_set
train_set = pd.concat([train_set, new_train_set])
tuning_data = tune_set
if use_test_data:
    test_data = test_set

train_data = train_set

train_data = TabularDataset(train_data)
tuning_data = TabularDataset(tuning_data)

if use_test_data:
    test_data = TabularDataset(test_data)

```

4 Modeling

```

[10]: import os

# if submissions folder does not exist, create it
if not os.path.exists('submissions'):
    os.makedirs('submissions')

# Get the last submission number
last_submission_number = int(max([int(filename.split('_')[1].split('.')[0]) for
    ↪ filename in os.listdir('submissions') if "submission" in filename]))
print("Last submission number:", last_submission_number)
print("Now creating submission number:", last_submission_number + 1)

# Create the new filename
new_filename = f'submission_{last_submission_number + 1}'

print("New filename:", new_filename)

```

```

Last submission number: 135
Now creating submission number: 136
New filename: submission_136

```

```

[11]: predictors = [None, None, None]

```

```

[12]: def fit_predictor_for_location(loc):
    # All of these hyperparameters have been found by experimenting with some
    ↪ standard parameters in AutoGluon, and then only using the best ones for each
    ↪ location to make the train time shorter

```

```

    r118 = {'extra_trees': True, 'feature_fraction': 0.7832570544199176,
    ↪ 'learning_rate': 0.021720607471727896, 'min_data_in_leaf': 3, 'num_leaves':
    ↪ 21, 'ag_args': {'name_suffix': '_r118', 'priority': 17}}
    r51 = {'bs': 1024, 'emb_drop': 0.6046989241462619, 'epochs': 48, 'layers':
    ↪ [200, 100, 50], 'lr': 0.00775309042164966, 'ps': 0.09244767444160731,
    ↪ 'ag_args': {'name_suffix': '_r51', 'priority': 12}}
    r145 = {'bs': 128, 'emb_drop': 0.44339037504795686, 'epochs': 31, 'layers':
    ↪ [400, 200, 100], 'lr': 0.008615195908919904, 'ps': 0.19220253419114286,
    ↪ 'ag_args': {'name_suffix': '_r145', 'priority': 9}}
    lgbmXT = {'extra_trees': True, 'ag_args': {'name_suffix': 'XT'}}
    if loc == "A":
        hyperparameters = {
            'NN_TORCH': {},
            'GBM': [lgbmXT, r118, 'GBMLarge'], #, r_118],
            'FASTAI': [r51],
        }
    elif loc == "B":
        hyperparameters = {
            'NN_TORCH': {},
            'GBM': [r118],
            'FASTAI': [{}, r145],
        }

    elif loc == "C":
        hyperparameters = {
            'NN_TORCH': {},
            'KNN': [{'weights': 'uniform', 'ag_args': {'name_suffix':
    ↪ 'Unif'}}}],
            'GBM': [r118],
            'FASTAI': [r51],
        }

    predictor = TabularPredictor(
        label=label,
        eval_metric=metric,
        path=f"AutogluonModels/{new_filename}_{loc}",
    ).fit(
        train_data=train_data[train_data["location"] == loc].
    ↪ drop(columns=["ds"]),
        time_limit=time_limit,
        presets=presets,
        num_stack_levels=num_stack_levels,
        num_bag_folds=num_bag_folds,
        num_bag_sets=num_bag_sets,
        tuning_data=tuning_data[tuning_data["location"] == loc].
    ↪ reset_index(drop=True).drop(columns=["ds"]),

```

```

        hyperparameters = hyperparameters,
        use_bag_holdout=use_bag_holdout,
        #excluded_model_types=excluded_model_types
    )

    # evaluate on test data
    if use_test_data:
        t = test_data[test_data["location"] == loc]
        perf = predictor.evaluate(t)
        print("Evaluation on test data:")
        print(perf[predictor.eval_metric.name])

    return predictor

loc = "A"
predictors[0] = fit_predictor_for_location(loc)

```

Presets specified: ['experimental_zeroshot_hpo_hybrid']
Stack configuration (auto_stack=True): num_stack_levels=0, num_bag_folds=8, num_bag_sets=3
Beginning AutoGluon training ...
AutoGluon will save models to "AutogluonModels/submission_136_A/"
AutoGluon Version: 0.8.2
Python Version: 3.10.12
Operating System: Darwin
Platform Machine: arm64
Platform Version: Darwin Kernel Version 22.1.0: Sun Oct 9 20:15:09 PDT 2022; root:xnu-8792.41.9~2/RELEASE_ARM64_T6000
Disk Space Avail: 99.51 GB / 494.38 GB (20.1%)
Train Data Rows: 30934
Train Data Columns: 44
Tuning Data Rows: 1485
Tuning Data Columns: 44
Label Column: y
Preprocessing data ...
AutoGluon infers your prediction problem is: 'regression' (because dtype of label-column == float and many unique label-values observed).
Label info (max, min, mean, stddev): (5733.42, 0.0, 673.41535, 1195.24)
If 'regression' is not the correct problem_type, please manually specify the problem_type parameter during predictor init (You may specify problem_type as one of: ['binary', 'multiclass', 'regression'])
Using Feature Generators to preprocess the data ...
Fitting AutoMLPipelineFeatureGenerator...
Available Memory: 3331.4 MB
Train Data (Original) Memory Usage: 13.03 MB (0.4% of available memory)
Inferring data type of each feature based on column values. Set feature_metadata_in to manually specify special dtypes of the features.
Stage 1 Generators:

```

        Fitting AsTypeFeatureGenerator...
        Note: Converting 2 features to boolean dtype as they
only contain 2 unique values.
    Stage 2 Generators:
        Fitting FillNaFeatureGenerator...
    Stage 3 Generators:
        Fitting IdentityFeatureGenerator...
    Stage 4 Generators:
        Fitting DropUniqueFeatureGenerator...
    Stage 5 Generators:
        Fitting DropDuplicatesFeatureGenerator...
    Useless Original Features (Count: 3): ['elevation:m', 'snow_drift:idx',
'location']

        These features carry no predictive signal and should be manually
investigated.

        This is typically a feature which has the same value for all
rows.

        These features do not need to be present at inference time.
    Types of features in original data (raw dtype, special dtypes):
        ('float', []) : 40 | ['absolute_humidity_2m:gm3',
'air_density_2m:kgm3', 'ceiling_height_agl:m', 'clear_sky_energy_1h:J',
'clear_sky_rad:W', ...]
        ('int', [])   : 1 | ['is_estimated']
    Types of features in processed data (raw dtype, special dtypes):
        ('float', []) : 39 | ['absolute_humidity_2m:gm3',
'air_density_2m:kgm3', 'ceiling_height_agl:m', 'clear_sky_energy_1h:J',
'clear_sky_rad:W', ...]
        ('int', ['bool']) : 2 | ['snow_density:kgm3', 'is_estimated']
    0.1s = Fit runtime
    41 features in original data used to generate 41 features in processed
data.

    Train Data (Processed) Memory Usage: 10.18 MB (0.3% of available memory)
Data preprocessing and feature engineering runtime = 0.11s ...
AutoGluon will gauge predictive performance using evaluation metric:
'mean_absolute_error'

    This metric's sign has been flipped to adhere to being higher_is_better.
The metric score can be multiplied by -1 to get the metric value.

    To change this, specify the eval_metric parameter of Predictor()
use_bag_holdout=True, will use tuning_data as holdout (will not be used for
early stopping).
User-specified model hyperparameters to be fit:
{
    'NN_TORCH': {},
    'GBM': [{'extra_trees': True, 'ag_args': {'name_suffix': 'XT'}},
{'extra_trees': True, 'feature_fraction': 0.7832570544199176, 'learning_rate':
0.021720607471727896, 'min_data_in_leaf': 3, 'num_leaves': 21, 'ag_args':
{'name_suffix': '_r118', 'priority': 17}}, 'GBMLarge'],
    'FASTAI': [{'bs': 1024, 'emb_drop': 0.6046989241462619, 'epochs': 48,

```



```
'layers': [200, 100, 50], 'lr': 0.00775309042164966, 'ps': 0.09244767444160731,
'ag_args': {'name_suffix': '_r51', 'priority': 12}}],
}
```

Fitting 5 L1 models ...

Fitting model: LightGBMXT_BAG_L1 ...

Will use sequential fold fitting strategy because import of ray failed. Reason:
ray is required to train folds in parallel. A quick tip is to install via `pip
install ray==2.2.0`, or use sequential fold fitting by passing
`sequential_local` to `ag_args_ensemble` when calling tabular.fitFor example:
`predictor.fit(..., ag_args_ensemble={'fold_fitting_strategy':
'sequential_local'})`

Fitting 24 child models (S1F1 - S3F8) | Fitting with
SequentialLocalFoldFittingStrategy

```
[1000] valid_set's l1: 188.903
[2000] valid_set's l1: 183.38
[3000] valid_set's l1: 180.745
[4000] valid_set's l1: 178.855
[5000] valid_set's l1: 177.036
[6000] valid_set's l1: 176.178
[7000] valid_set's l1: 175.372
[8000] valid_set's l1: 174.791
[9000] valid_set's l1: 174.317
[10000] valid_set's l1: 173.968
[1000] valid_set's l1: 195.036
[2000] valid_set's l1: 190.396
[3000] valid_set's l1: 187.398
[4000] valid_set's l1: 185.562
[5000] valid_set's l1: 184.402
[6000] valid_set's l1: 183.498
[7000] valid_set's l1: 182.975
[8000] valid_set's l1: 182.6
[9000] valid_set's l1: 182.308
[10000] valid_set's l1: 182.089
[1000] valid_set's l1: 174.756
[2000] valid_set's l1: 170.021
[3000] valid_set's l1: 167.451
[4000] valid_set's l1: 166.491
[5000] valid_set's l1: 165.668
[6000] valid_set's l1: 164.958
[7000] valid_set's l1: 164.495
[8000] valid_set's l1: 164.319
[9000] valid_set's l1: 164.064
[10000] valid_set's l1: 163.91
[1000] valid_set's l1: 185.905
[2000] valid_set's l1: 180.669
[3000] valid_set's l1: 178.641
[4000] valid_set's l1: 177.158
```

[5000] valid_set's l1: 176.004
[6000] valid_set's l1: 175.196
[7000] valid_set's l1: 174.998
[8000] valid_set's l1: 174.683
[9000] valid_set's l1: 174.314
[10000] valid_set's l1: 174.04
[1000] valid_set's l1: 183.164
[2000] valid_set's l1: 176.218
[3000] valid_set's l1: 172.938
[4000] valid_set's l1: 171.415
[5000] valid_set's l1: 169.986
[6000] valid_set's l1: 169.039
[7000] valid_set's l1: 168.349
[8000] valid_set's l1: 167.719
[9000] valid_set's l1: 167.298
[10000] valid_set's l1: 166.983
[1000] valid_set's l1: 170.514
[2000] valid_set's l1: 165.088
[3000] valid_set's l1: 162.919
[4000] valid_set's l1: 161.49
[5000] valid_set's l1: 160.782
[6000] valid_set's l1: 160.182
[7000] valid_set's l1: 159.752
[8000] valid_set's l1: 159.568
[9000] valid_set's l1: 159.34
[10000] valid_set's l1: 159.161
[1000] valid_set's l1: 189.562
[2000] valid_set's l1: 183.203
[3000] valid_set's l1: 180.322
[4000] valid_set's l1: 178.848
[5000] valid_set's l1: 178.163
[6000] valid_set's l1: 177.178
[7000] valid_set's l1: 176.468
[8000] valid_set's l1: 175.848
[9000] valid_set's l1: 175.54
[10000] valid_set's l1: 175.128
[1000] valid_set's l1: 182.64
[2000] valid_set's l1: 176.705
[3000] valid_set's l1: 173.867
[4000] valid_set's l1: 171.977
[5000] valid_set's l1: 170.847
[6000] valid_set's l1: 170.152
[7000] valid_set's l1: 169.313
[8000] valid_set's l1: 168.821
[9000] valid_set's l1: 168.33
[10000] valid_set's l1: 167.86
[1000] valid_set's l1: 178.947
[2000] valid_set's l1: 172.472

[3000] valid_set's l1: 169.38
[4000] valid_set's l1: 167.535
[5000] valid_set's l1: 166.495
[6000] valid_set's l1: 165.844
[7000] valid_set's l1: 165.265
[8000] valid_set's l1: 164.826
[9000] valid_set's l1: 164.559
[10000] valid_set's l1: 164.303
[1000] valid_set's l1: 185.045
[2000] valid_set's l1: 179.975
[3000] valid_set's l1: 177.439
[4000] valid_set's l1: 176.165
[5000] valid_set's l1: 174.805
[6000] valid_set's l1: 174.191
[7000] valid_set's l1: 173.858
[8000] valid_set's l1: 173.428
[9000] valid_set's l1: 173.314
[10000] valid_set's l1: 173.141
[1000] valid_set's l1: 191.245
[2000] valid_set's l1: 184.717
[3000] valid_set's l1: 181.976
[4000] valid_set's l1: 180.495
[5000] valid_set's l1: 179.241
[6000] valid_set's l1: 178.166
[7000] valid_set's l1: 177.725
[8000] valid_set's l1: 177.246
[9000] valid_set's l1: 176.968
[10000] valid_set's l1: 176.726
[1000] valid_set's l1: 182.476
[2000] valid_set's l1: 175.709
[3000] valid_set's l1: 172.75
[4000] valid_set's l1: 171.057
[5000] valid_set's l1: 170.108
[6000] valid_set's l1: 169.455
[7000] valid_set's l1: 169
[8000] valid_set's l1: 168.689
[9000] valid_set's l1: 168.406
[10000] valid_set's l1: 168.183
[1000] valid_set's l1: 184.457
[2000] valid_set's l1: 178.408
[3000] valid_set's l1: 175.886
[4000] valid_set's l1: 174.314
[5000] valid_set's l1: 172.878
[6000] valid_set's l1: 171.736
[7000] valid_set's l1: 170.937
[8000] valid_set's l1: 170.45
[9000] valid_set's l1: 169.89
[10000] valid_set's l1: 169.539

[1000] valid_set's l1: 180.593
[2000] valid_set's l1: 175.147
[3000] valid_set's l1: 172.469
[4000] valid_set's l1: 170.894
[5000] valid_set's l1: 169.911
[6000] valid_set's l1: 169.123
[7000] valid_set's l1: 168.504
[8000] valid_set's l1: 168.142
[9000] valid_set's l1: 167.828
[10000] valid_set's l1: 167.573
[1000] valid_set's l1: 180.812
[2000] valid_set's l1: 176.977
[3000] valid_set's l1: 175.571
[4000] valid_set's l1: 174.351
[5000] valid_set's l1: 173.793
[6000] valid_set's l1: 173.001
[7000] valid_set's l1: 172.304
[8000] valid_set's l1: 172.005
[9000] valid_set's l1: 171.752
[10000] valid_set's l1: 171.429
[1000] valid_set's l1: 183.231
[2000] valid_set's l1: 178.251
[3000] valid_set's l1: 176.468
[4000] valid_set's l1: 174.974
[5000] valid_set's l1: 174.559
[6000] valid_set's l1: 174.02
[7000] valid_set's l1: 173.677
[8000] valid_set's l1: 173.163
[9000] valid_set's l1: 172.943
[10000] valid_set's l1: 172.713
[1000] valid_set's l1: 180.573
[2000] valid_set's l1: 176.141
[3000] valid_set's l1: 174.435
[4000] valid_set's l1: 172.41
[5000] valid_set's l1: 171.164
[6000] valid_set's l1: 170.282
[7000] valid_set's l1: 169.755
[8000] valid_set's l1: 169.345
[9000] valid_set's l1: 169.034
[10000] valid_set's l1: 168.764
[1000] valid_set's l1: 180.9
[2000] valid_set's l1: 176.002
[3000] valid_set's l1: 173.366
[4000] valid_set's l1: 171.852
[5000] valid_set's l1: 170.949
[6000] valid_set's l1: 170.284
[7000] valid_set's l1: 169.924
[8000] valid_set's l1: 169.56

[9000] valid_set's l1: 169.216
[10000] valid_set's l1: 169.169
[1000] valid_set's l1: 169.315
[2000] valid_set's l1: 164.021
[3000] valid_set's l1: 161.538
[4000] valid_set's l1: 160.216
[5000] valid_set's l1: 159.259
[6000] valid_set's l1: 158.348
[7000] valid_set's l1: 157.875
[8000] valid_set's l1: 157.31
[9000] valid_set's l1: 157.147
[10000] valid_set's l1: 156.925
[1000] valid_set's l1: 188.974
[2000] valid_set's l1: 182.654
[3000] valid_set's l1: 179.72
[4000] valid_set's l1: 177.697
[5000] valid_set's l1: 175.948
[6000] valid_set's l1: 174.752
[7000] valid_set's l1: 174.017
[8000] valid_set's l1: 173.503
[9000] valid_set's l1: 173.113
[10000] valid_set's l1: 172.848
[1000] valid_set's l1: 177.641
[2000] valid_set's l1: 171.296
[3000] valid_set's l1: 169.291
[4000] valid_set's l1: 168.055
[5000] valid_set's l1: 167.098
[6000] valid_set's l1: 166.211
[7000] valid_set's l1: 165.897
[8000] valid_set's l1: 165.408
[9000] valid_set's l1: 165.057
[10000] valid_set's l1: 164.855
[1000] valid_set's l1: 186.754
[2000] valid_set's l1: 181.009
[3000] valid_set's l1: 178.294
[4000] valid_set's l1: 176.756
[5000] valid_set's l1: 175.996
[6000] valid_set's l1: 175.233
[7000] valid_set's l1: 174.687
[8000] valid_set's l1: 174.28
[9000] valid_set's l1: 174.137
[10000] valid_set's l1: 173.962
[1000] valid_set's l1: 194.188
[2000] valid_set's l1: 190.187
[3000] valid_set's l1: 187.925
[4000] valid_set's l1: 186.35
[5000] valid_set's l1: 185.104
[6000] valid_set's l1: 183.685

```

[7000] valid_set's l1: 182.919
[8000] valid_set's l1: 182.264
[9000] valid_set's l1: 181.979
[10000] valid_set's l1: 181.54
[1000] valid_set's l1: 190.42
[2000] valid_set's l1: 183.677
[3000] valid_set's l1: 181.484
[4000] valid_set's l1: 179.655
[5000] valid_set's l1: 178.61
[6000] valid_set's l1: 177.807
[7000] valid_set's l1: 177.081
[8000] valid_set's l1: 176.588
[9000] valid_set's l1: 176.192
[10000] valid_set's l1: 175.922

-86.158 = Validation score (-mean_absolute_error)
3506.39s = Training runtime
16.76s = Validation runtime
Fitting model: NeuralNetTorch_BAG_L1 ...
Fitting 24 child models (S1F1 - S3F8) | Fitting with
SequentialLocalFoldFittingStrategy
-87.4034 = Validation score (-mean_absolute_error)
738.13s = Training runtime
0.42s = Validation runtime
Fitting model: LightGBM_r118_BAG_L1 ...
Fitting 24 child models (S1F1 - S3F8) | Fitting with
SequentialLocalFoldFittingStrategy

[1000] valid_set's l1: 196.904
[2000] valid_set's l1: 190.607
[3000] valid_set's l1: 187.091
[4000] valid_set's l1: 184.168
[5000] valid_set's l1: 182.265
[6000] valid_set's l1: 180.313
[7000] valid_set's l1: 179.262
[8000] valid_set's l1: 178.294
[9000] valid_set's l1: 177.417
[10000] valid_set's l1: 176.686
[1000] valid_set's l1: 203.298
[2000] valid_set's l1: 196.232
[3000] valid_set's l1: 192.207
[4000] valid_set's l1: 189.201
[5000] valid_set's l1: 186.588
[6000] valid_set's l1: 184.734
[7000] valid_set's l1: 183.29
[8000] valid_set's l1: 182.073
[9000] valid_set's l1: 181.009
[10000] valid_set's l1: 180.123
[1000] valid_set's l1: 184.004

```

[2000] valid_set's l1: 177.114
[3000] valid_set's l1: 173.096
[4000] valid_set's l1: 170.176
[5000] valid_set's l1: 168.127
[6000] valid_set's l1: 166.408
[7000] valid_set's l1: 165.167
[8000] valid_set's l1: 164.149
[9000] valid_set's l1: 163.541
[10000] valid_set's l1: 163.015
[1000] valid_set's l1: 196.856
[2000] valid_set's l1: 190.083
[3000] valid_set's l1: 185.916
[4000] valid_set's l1: 183.064
[5000] valid_set's l1: 180.458
[6000] valid_set's l1: 178.905
[7000] valid_set's l1: 177.304
[8000] valid_set's l1: 176.191
[9000] valid_set's l1: 175.247
[10000] valid_set's l1: 174.466
[1000] valid_set's l1: 191.653
[2000] valid_set's l1: 185.251
[3000] valid_set's l1: 180.729
[4000] valid_set's l1: 177.725
[5000] valid_set's l1: 175.324
[6000] valid_set's l1: 173.372
[7000] valid_set's l1: 171.959
[8000] valid_set's l1: 170.595
[9000] valid_set's l1: 169.533
[10000] valid_set's l1: 168.745
[1000] valid_set's l1: 180.134
[2000] valid_set's l1: 173.747
[3000] valid_set's l1: 169.607
[4000] valid_set's l1: 166.973
[5000] valid_set's l1: 164.793
[6000] valid_set's l1: 163.125
[7000] valid_set's l1: 162.027
[8000] valid_set's l1: 160.871
[9000] valid_set's l1: 160.201
[10000] valid_set's l1: 159.58
[1000] valid_set's l1: 198.353
[2000] valid_set's l1: 191.907
[3000] valid_set's l1: 188.054
[4000] valid_set's l1: 185.566
[5000] valid_set's l1: 183.65
[6000] valid_set's l1: 181.883
[7000] valid_set's l1: 180.434
[8000] valid_set's l1: 179.533
[9000] valid_set's l1: 178.369

[10000] valid_set's l1: 177.757
[1000] valid_set's l1: 195.547
[2000] valid_set's l1: 187.908
[3000] valid_set's l1: 183.575
[4000] valid_set's l1: 180.677
[5000] valid_set's l1: 178.622
[6000] valid_set's l1: 177.35
[7000] valid_set's l1: 176.024
[8000] valid_set's l1: 175.023
[9000] valid_set's l1: 174.088
[10000] valid_set's l1: 173.247
[1000] valid_set's l1: 189.904
[2000] valid_set's l1: 183.13
[3000] valid_set's l1: 177.894
[4000] valid_set's l1: 174.506
[5000] valid_set's l1: 171.897
[6000] valid_set's l1: 170.295
[7000] valid_set's l1: 168.7
[8000] valid_set's l1: 167.37
[9000] valid_set's l1: 166.238
[10000] valid_set's l1: 165.322
[1000] valid_set's l1: 197.213
[2000] valid_set's l1: 189.562
[3000] valid_set's l1: 185.405
[4000] valid_set's l1: 182.243
[5000] valid_set's l1: 180.161
[6000] valid_set's l1: 178.32
[7000] valid_set's l1: 177.122
[8000] valid_set's l1: 176.008
[9000] valid_set's l1: 175.399
[10000] valid_set's l1: 174.577
[1000] valid_set's l1: 204.181
[2000] valid_set's l1: 196.879
[3000] valid_set's l1: 192.344
[4000] valid_set's l1: 189.22
[5000] valid_set's l1: 186.548
[6000] valid_set's l1: 184.907
[7000] valid_set's l1: 183.32
[8000] valid_set's l1: 182.064
[9000] valid_set's l1: 181.097
[10000] valid_set's l1: 180.268
[1000] valid_set's l1: 192.529
[2000] valid_set's l1: 186.319
[3000] valid_set's l1: 181.962
[4000] valid_set's l1: 178.822
[5000] valid_set's l1: 176.509
[6000] valid_set's l1: 174.886
[7000] valid_set's l1: 173.391

[8000] valid_set's l1: 172.154
[9000] valid_set's l1: 171.248
[10000] valid_set's l1: 170.6
[1000] valid_set's l1: 191.938
[2000] valid_set's l1: 185.59
[3000] valid_set's l1: 181.1
[4000] valid_set's l1: 178.076
[5000] valid_set's l1: 176.015
[6000] valid_set's l1: 174.44
[7000] valid_set's l1: 172.91
[8000] valid_set's l1: 171.768
[9000] valid_set's l1: 170.842
[10000] valid_set's l1: 169.949
[1000] valid_set's l1: 189.056
[2000] valid_set's l1: 183.466
[3000] valid_set's l1: 179.977
[4000] valid_set's l1: 177.576
[5000] valid_set's l1: 175.39
[6000] valid_set's l1: 173.471
[7000] valid_set's l1: 172.055
[8000] valid_set's l1: 170.613
[9000] valid_set's l1: 169.76
[10000] valid_set's l1: 168.877
[1000] valid_set's l1: 188.691
[2000] valid_set's l1: 182.749
[3000] valid_set's l1: 179.235
[4000] valid_set's l1: 176.948
[5000] valid_set's l1: 175.476
[6000] valid_set's l1: 173.839
[7000] valid_set's l1: 172.825
[8000] valid_set's l1: 172.013
[9000] valid_set's l1: 171.422
[10000] valid_set's l1: 170.759
[1000] valid_set's l1: 194.373
[2000] valid_set's l1: 187.312
[3000] valid_set's l1: 182.907
[4000] valid_set's l1: 180.125
[5000] valid_set's l1: 178.03
[6000] valid_set's l1: 176.397
[7000] valid_set's l1: 174.834
[8000] valid_set's l1: 174.034
[9000] valid_set's l1: 173.403
[10000] valid_set's l1: 172.656
[1000] valid_set's l1: 188.004
[2000] valid_set's l1: 182.057
[3000] valid_set's l1: 178.652
[4000] valid_set's l1: 176.219
[5000] valid_set's l1: 174.01

[6000] valid_set's l1: 172.537
[7000] valid_set's l1: 171.185
[8000] valid_set's l1: 170.236
[9000] valid_set's l1: 169.503
[10000] valid_set's l1: 168.7
[1000] valid_set's l1: 191.854
[2000] valid_set's l1: 184.968
[3000] valid_set's l1: 180.693
[4000] valid_set's l1: 177.455
[5000] valid_set's l1: 175.152
[6000] valid_set's l1: 173.38
[7000] valid_set's l1: 172.019
[8000] valid_set's l1: 170.966
[9000] valid_set's l1: 170.055
[10000] valid_set's l1: 169.255
[1000] valid_set's l1: 177.718
[2000] valid_set's l1: 171.726
[3000] valid_set's l1: 167.898
[4000] valid_set's l1: 165.41
[5000] valid_set's l1: 163.324
[6000] valid_set's l1: 161.618
[7000] valid_set's l1: 160.321
[8000] valid_set's l1: 159.266
[9000] valid_set's l1: 158.491
[10000] valid_set's l1: 157.751
[1000] valid_set's l1: 198.626
[2000] valid_set's l1: 192.688
[3000] valid_set's l1: 188.836
[4000] valid_set's l1: 185.85
[5000] valid_set's l1: 183.374
[6000] valid_set's l1: 181.548
[7000] valid_set's l1: 180.115
[8000] valid_set's l1: 179.12
[9000] valid_set's l1: 178.118
[10000] valid_set's l1: 177.145
[1000] valid_set's l1: 187.4
[2000] valid_set's l1: 179.602
[3000] valid_set's l1: 175.374
[4000] valid_set's l1: 172.522
[5000] valid_set's l1: 170.574
[6000] valid_set's l1: 168.967
[7000] valid_set's l1: 167.568
[8000] valid_set's l1: 166.383
[9000] valid_set's l1: 165.547
[10000] valid_set's l1: 164.944
[1000] valid_set's l1: 197.735
[2000] valid_set's l1: 190.365
[3000] valid_set's l1: 185.974

```

[4000] valid_set's l1: 183.087
[5000] valid_set's l1: 180.511
[6000] valid_set's l1: 178.762
[7000] valid_set's l1: 177.14
[8000] valid_set's l1: 175.715
[9000] valid_set's l1: 174.57
[10000] valid_set's l1: 173.814
[1000] valid_set's l1: 204.953
[2000] valid_set's l1: 197.709
[3000] valid_set's l1: 193.633
[4000] valid_set's l1: 190.905
[5000] valid_set's l1: 188.764
[6000] valid_set's l1: 187.202
[7000] valid_set's l1: 185.941
[8000] valid_set's l1: 184.871
[9000] valid_set's l1: 183.782
[10000] valid_set's l1: 182.961
[1000] valid_set's l1: 200.158
[2000] valid_set's l1: 193.101
[3000] valid_set's l1: 188.776
[4000] valid_set's l1: 185.75
[5000] valid_set's l1: 183.555
[6000] valid_set's l1: 181.618
[7000] valid_set's l1: 180.183
[8000] valid_set's l1: 178.781
[9000] valid_set's l1: 177.689
[10000] valid_set's l1: 176.924

-85.2617          = Validation score  (-mean_absolute_error)
1744.63s          = Training  runtime
11.4s            = Validation runtime
Fitting model: NeuralNetFastAI_r51_BAG_L1 ...
Fitting 24 child models (S1F1 - S3F8) | Fitting with
SequentialLocalFoldFittingStrategy
-95.69           = Validation score  (-mean_absolute_error)
273.05s          = Training  runtime
0.31s           = Validation runtime
Fitting model: LightGBMLarge_BAG_L1 ...
Fitting 24 child models (S1F1 - S3F8) | Fitting with
SequentialLocalFoldFittingStrategy

[1000] valid_set's l1: 180.725
[2000] valid_set's l1: 179.262
[3000] valid_set's l1: 178.983
[4000] valid_set's l1: 178.905
[5000] valid_set's l1: 178.876
[6000] valid_set's l1: 178.871
[7000] valid_set's l1: 178.868
[8000] valid_set's l1: 178.867

```

```
[9000] valid_set's l1: 178.866
[10000] valid_set's l1: 178.866
```

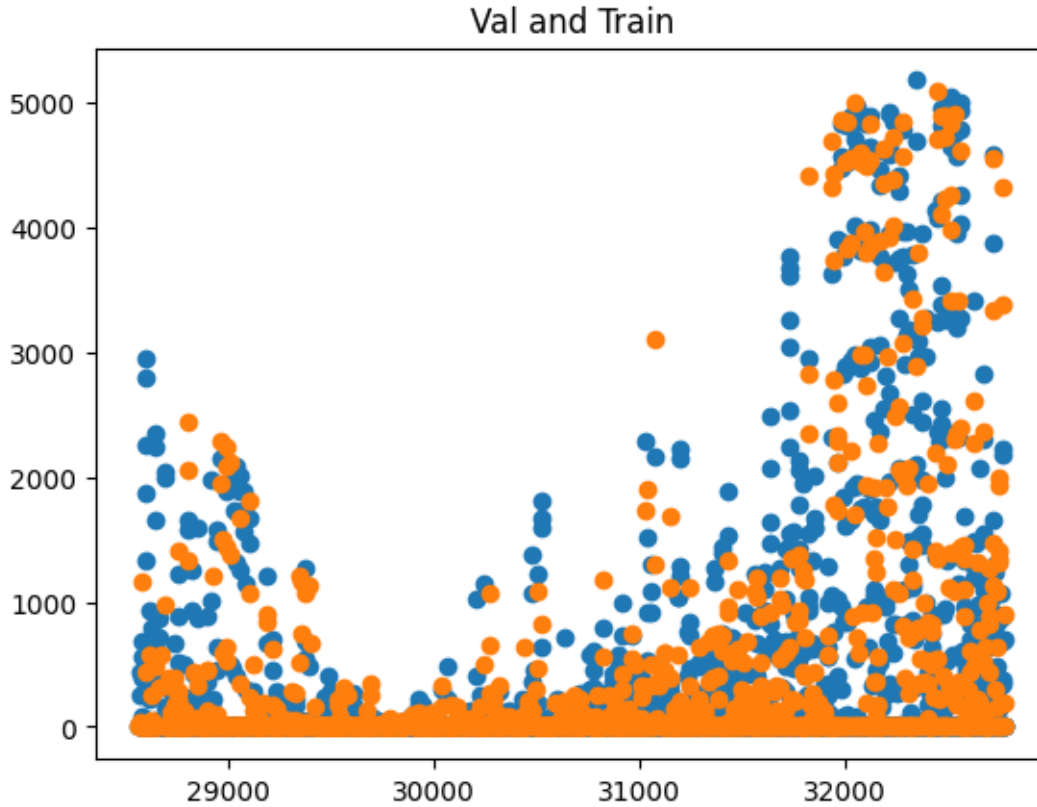
```
[ ]: import matplotlib.pyplot as plt
leaderboards = [None, None, None]
def leaderboard_for_location(i, loc):
    plt.scatter(train_data[(train_data["location"] == loc) &
    ↪(train_data["is_estimated"]==True)]["y"].index,
    ↪train_data[(train_data["location"] == loc) &
    ↪(train_data["is_estimated"]==True)]["y"])
    plt.scatter(tuning_data[tuning_data["location"] == loc]["y"].index,
    ↪tuning_data[tuning_data["location"] == loc]["y"])
    plt.title("Val and Train")
    plt.show()

    if use_test_data:
        lb = predictors[i].leaderboard(test_data[test_data["location"] == loc])
        lb["location"] = loc
        plt.scatter(test_data[test_data["location"] == loc]["y"].index,
    ↪test_data[test_data["location"] == loc]["y"])
        plt.title("Test")

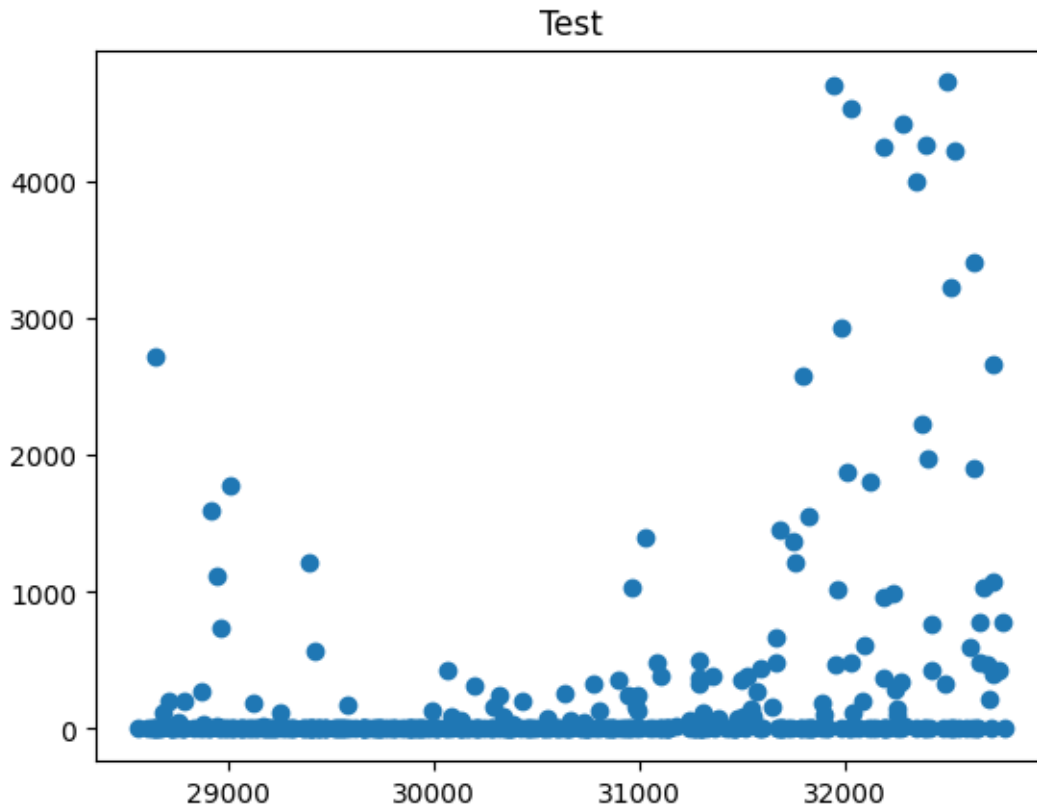
    return lb

return pd.DataFrame()

leaderboards[0] = leaderboard_for_location(0, loc)
```



	model	score_test	score_val	pred_time_test
pred_time_val	fit_time	pred_time_test_marginal	pred_time_val_marginal	
fit_time_marginal	stack_level	can_infer	fit_order	
0	LightGBMXT_BAG_L1	-105.629299	-86.171125	2.757117
267.977811	66.673744		2.757117	267.977811
66.673744	1	True	1	
1	WeightedEnsemble_L2	-105.742732	-82.247352	15.061079
835.773022	458.526718		0.004030	0.000701
0.214789	2	True	6	
2	NeuralNetFastAI_r51_BAG_L1	-106.247964	-95.666327	0.383407
0.937461	38.080469		0.383407	0.937461
38.080469	1	True	4	
3	LightGBM_r118_BAG_L1	-106.870624	-85.259916	2.342607
188.234109	51.846491		2.342607	188.234109
51.846491	1	True	3	
4	NeuralNetTorch_BAG_L1	-112.454180	-86.980762	0.562749
1.326716	136.740541		0.562749	1.326716
136.740541	1	True	2	
5	LightGBMLarge_BAG_L1	-117.513584	-87.298688	9.394577
378.233686	203.051153		9.394577	378.233686
203.051153	1	True	5	



```
[ ]: loc = "B"
predictors[1] = fit_predictor_for_location(loc)
leaderboards[1] = leaderboard_for_location(1, loc)
```

```
Presets specified: ['experimental_zeroshot_hpo_hybrid']
Stack configuration (auto_stack=True): num_stack_levels=0, num_bag_folds=8,
num_bag_sets=3
Beginning AutoGluon training ...
AutoGluon will save models to "AutogluonModels/submission_142_B/"
AutoGluon Version: 0.8.2
Python Version: 3.10.12
Operating System: Linux
Platform Machine: x86_64
Platform Version: #1 SMP Debian 5.10.197-1 (2023-09-29)
Disk Space Avail: 137.93 GB / 315.93 GB (43.7%)
Train Data Rows: 27377
Train Data Columns: 44
Tuning Data Rows: 1485
Tuning Data Columns: 44
Label Column: y
Preprocessing data ...
```

```

AutoGluon infers your prediction problem is: 'regression' (because dtype of
label-column == float and many unique label-values observed).
    Label info (max, min, mean, stddev): (1152.3, -0.0, 98.11625, 206.48535)
    If 'regression' is not the correct problem_type, please manually specify
the problem_type parameter during predictor init (You may specify problem_type
as one of: ['binary', 'multiclass', 'regression'])
Using Feature Generators to preprocess the data ...
Fitting AutoMLPipelineFeatureGenerator...
    Available Memory:                131221.19 MB
    Train Data (Original) Memory Usage: 11.6 MB (0.0% of available memory)
    Inferring data type of each feature based on column values. Set
feature_metadata_in to manually specify special dtypes of the features.
    Stage 1 Generators:
        Fitting AsTypeFeatureGenerator...
            Note: Converting 2 features to boolean dtype as they
only contain 2 unique values.
    Stage 2 Generators:
        Fitting FillNaFeatureGenerator...
    Stage 3 Generators:
        Fitting IdentityFeatureGenerator...
    Stage 4 Generators:
        Fitting DropUniqueFeatureGenerator...
    Stage 5 Generators:
        Fitting DropDuplicatesFeatureGenerator...
    Useless Original Features (Count: 2): ['elevation:m', 'location']
        These features carry no predictive signal and should be manually
investigated.
            This is typically a feature which has the same value for all
rows.
                These features do not need to be present at inference time.
    Types of features in original data (raw dtype, special dtypes):
        ('float', []) : 41 | ['absolute_humidity_2m:gm3',
'air_density_2m:kgm3', 'ceiling_height_agl:m', 'clear_sky_energy_1h:J',
'clear_sky_rad:W', ...]
        ('int', []) : 1 | ['is_estimated']
    Types of features in processed data (raw dtype, special dtypes):
        ('float', []) : 40 | ['absolute_humidity_2m:gm3',
'air_density_2m:kgm3', 'ceiling_height_agl:m', 'clear_sky_energy_1h:J',
'clear_sky_rad:W', ...]
        ('int', ['bool']) : 2 | ['snow_density:kgm3', 'is_estimated']
    0.2s = Fit runtime
    42 features in original data used to generate 42 features in processed
data.
    Train Data (Processed) Memory Usage: 9.29 MB (0.0% of available memory)
Data preprocessing and feature engineering runtime = 0.22s ...
AutoGluon will gauge predictive performance using evaluation metric:
'mean_absolute_error'
    This metric's sign has been flipped to adhere to being higher_is_better.

```

The metric score can be multiplied by -1 to get the metric value.

To change this, specify the `eval_metric` parameter of `Predictor()` `use_bag_holdout=True`, will use `tuning_data` as holdout (will not be used for early stopping).

User-specified model hyperparameters to be fit:

```
{
    'NN_TORCH': {},
    'GBM': [{'extra_trees': True, 'feature_fraction': 0.7832570544199176,
'learning_rate': 0.021720607471727896, 'min_data_in_leaf': 3, 'num_leaves': 21,
'ag_args': {'name_suffix': '_r118', 'priority': 17}}],
    'FASTAI': [{}, {'bs': 128, 'emb_drop': 0.44339037504795686, 'epochs':
31, 'layers': [400, 200, 100], 'lr': 0.008615195908919904, 'ps':
0.19220253419114286, 'ag_args': {'name_suffix': '_r145', 'priority': 9}}],
}
```

Fitting 4 L1 models ...

Fitting model: NeuralNetFastAI_BAG_L1 ...

```
Fitting 24 child models (S1F1 - S3F8) | Fitting with
ParallelLocalFoldFittingStrategy
-13.1987      = Validation score    (-mean_absolute_error)
44.27s       = Training    runtime
1.55s        = Validation runtime
```

Fitting model: NeuralNetTorch_BAG_L1 ...

```
Fitting 24 child models (S1F1 - S3F8) | Fitting with
ParallelLocalFoldFittingStrategy
-12.8367      = Validation score    (-mean_absolute_error)
168.14s      = Training    runtime
1.15s        = Validation runtime
```

Fitting model: LightGBM_r118_BAG_L1 ...

```
Fitting 24 child models (S1F1 - S3F8) | Fitting with
ParallelLocalFoldFittingStrategy
-12.9212      = Validation score    (-mean_absolute_error)
47.36s       = Training    runtime
156.67s      = Validation runtime
```

Fitting model: NeuralNetFastAI_r145_BAG_L1 ...

```
Fitting 24 child models (S1F1 - S3F8) | Fitting with
ParallelLocalFoldFittingStrategy
-13.212       = Validation score    (-mean_absolute_error)
101.91s      = Training    runtime
3.09s        = Validation runtime
```

Fitting model: WeightedEnsemble_L2 ...

```
-12.3242      = Validation score    (-mean_absolute_error)
0.17s        = Training    runtime
0.0s         = Validation runtime
```

AutoGluon training complete, total runtime = 383.79s ... Best model:

"WeightedEnsemble_L2"

TabularPredictor saved. To load, use: `predictor =`

`TabularPredictor.load("AutogluonModels/submission_142_B/")`

Evaluation: `mean_absolute_error` on test data: -10.7894783641021

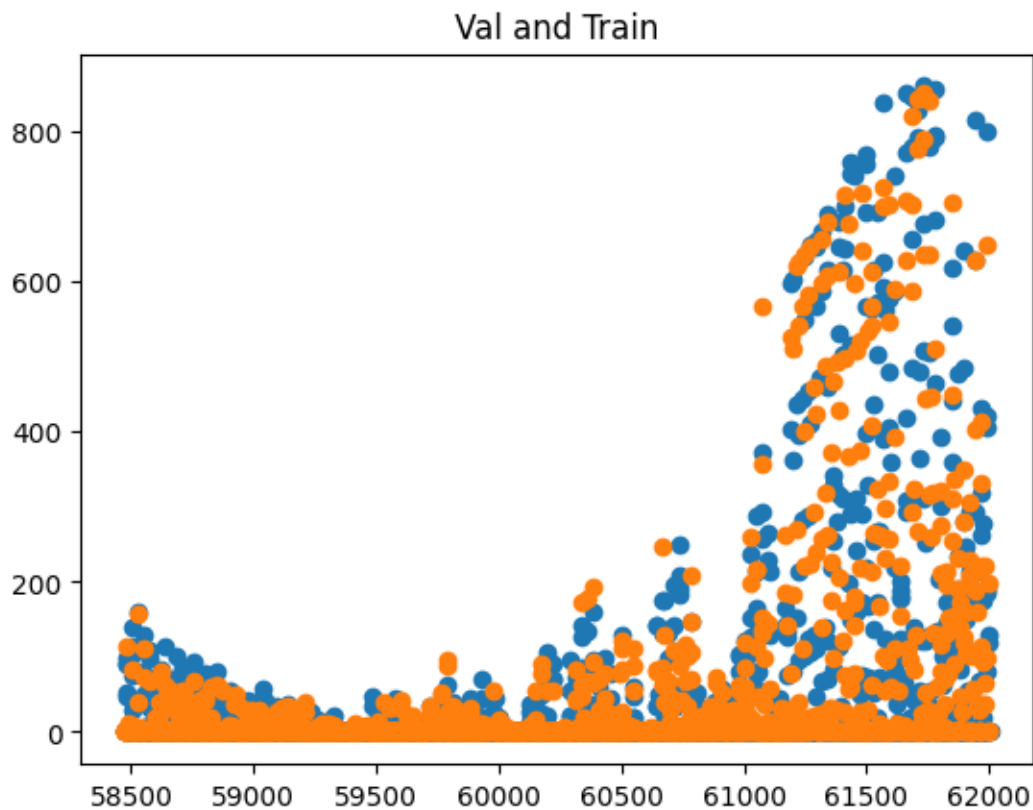
Note: Scores are always higher_is_better. This metric score can be multiplied by -1 to get the metric value.

Evaluations on test data:

```
{
  "mean_absolute_error": -10.7894783641021,
  "root_mean_squared_error": -29.37331033202028,
  "mean_squared_error": -862.7913598611693,
  "r2": 0.9628559948926422,
  "pearsonr": 0.9812682373560799,
  "median_absolute_error": -0.6040375232696533
}
```

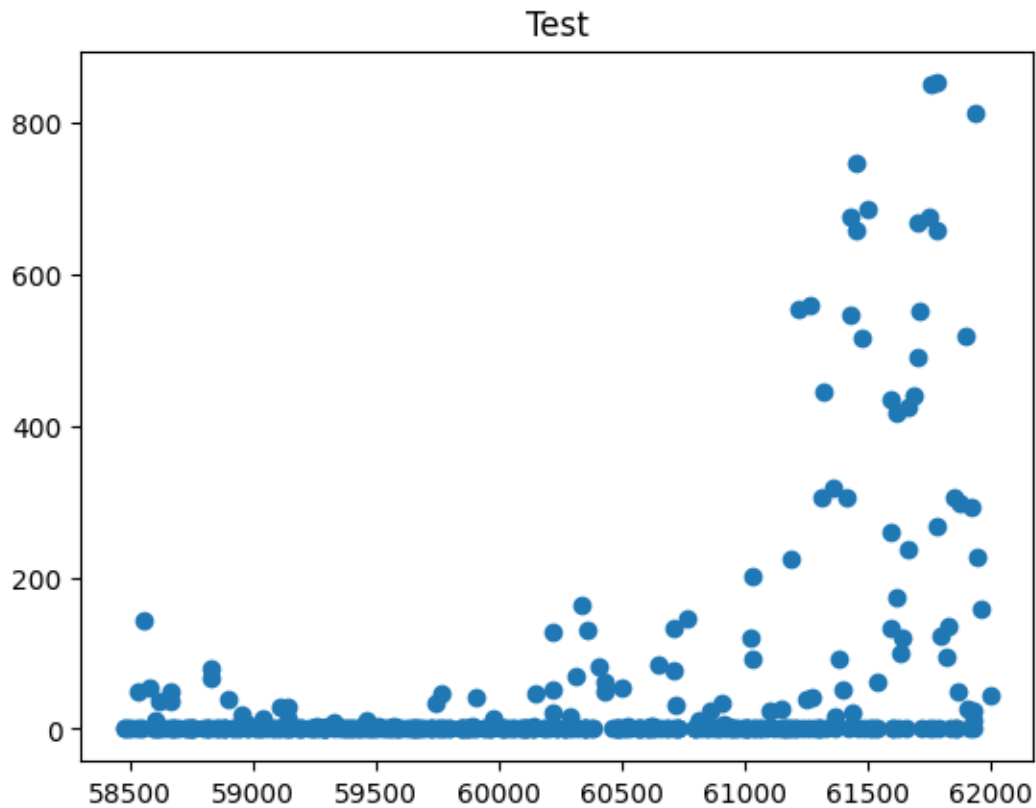
Evaluation on test data:

-10.7894783641021



	model	score_test	score_val	pred_time_test
0	WeightedEnsemble_L2	-10.789478	-12.324207	4.397277
162.457627	361.852653	0.003336	0.000642	
0.168811	2	True	5	
1	LightGBM_r118_BAG_L1	-11.008035	-12.921242	2.348640

156.671800	47.358682		2.348640	156.671800
47.358682	1	True	3	
2	NeuralNetTorch_BAG_L1	-11.182017	-12.836698	0.574365
1.147147	168.137105		0.574365	1.147147
168.137105	1	True	2	
3	NeuralNetFastAI_BAG_L1	-12.273773	-13.198685	0.498963
1.545790	44.273498		0.498963	1.545790
44.273498	1	True	1	
4	NeuralNetFastAI_r145_BAG_L1	-12.624919	-13.212044	0.971974
3.092247	101.914558		0.971974	3.092247
101.914558	1	True	4	



```
[ ]: loc = "C"
predictors[2] = fit_predictor_for_location(loc)
leaderboards[2] = leaderboard_for_location(2, loc)
```

Presets specified: ['experimental_zeroshot_hpo_hybrid']
Stack configuration (auto_stack=True): num_stack_levels=0, num_bag_folds=8,
num_bag_sets=3
Beginning AutoGluon training ...
AutoGluon will save models to "AutogluonModels/submission_142_C/"
AutoGluon Version: 0.8.2

```

Python Version:      3.10.12
Operating System:    Linux
Platform Machine:    x86_64
Platform Version:    #1 SMP Debian 5.10.197-1 (2023-09-29)
Disk Space Avail:    137.39 GB / 315.93 GB (43.5%)
Train Data Rows:     24073
Train Data Columns:  44
Tuning Data Rows:    1481
Tuning Data Columns: 44
Label Column: y
Preprocessing data ...
AutoGluon infers your prediction problem is: 'regression' (because dtype of
label-column == float and label-values can't be converted to int).
    Label info (max, min, mean, stddev): (999.6, -0.0, 80.87539, 169.67845)
    If 'regression' is not the correct problem_type, please manually specify
the problem_type parameter during predictor init (You may specify problem_type
as one of: ['binary', 'multiclass', 'regression'])
Using Feature Generators to preprocess the data ...
Fitting AutoMLPipelineFeatureGenerator...
    Available Memory:      131122.99 MB
    Train Data (Original) Memory Usage: 10.27 MB (0.0% of available memory)
    Inferring data type of each feature based on column values. Set
feature_metadata_in to manually specify special dtypes of the features.
    Stage 1 Generators:
        Fitting AsTypeFeatureGenerator...
            Note: Converting 2 features to boolean dtype as they
only contain 2 unique values.
    Stage 2 Generators:
        Fitting FillNaFeatureGenerator...
    Stage 3 Generators:
        Fitting IdentityFeatureGenerator...
    Stage 4 Generators:
        Fitting DropUniqueFeatureGenerator...
    Stage 5 Generators:
        Fitting DropDuplicatesFeatureGenerator...
    Useless Original Features (Count: 3): ['elevation:m', 'snow_drift:idx',
'location']
        These features carry no predictive signal and should be manually
investigated.
        This is typically a feature which has the same value for all
rows.
        These features do not need to be present at inference time.
    Types of features in original data (raw dtype, special dtypes):
        ('float', []) : 40 | ['absolute_humidity_2m:gm3',
'air_density_2m:kgm3', 'ceiling_height_agl:m', 'clear_sky_energy_1h:J',
'clear_sky_rad:W', ...]
        ('int', [])   : 1 | ['is_estimated']
    Types of features in processed data (raw dtype, special dtypes):

```

```

('float', []) : 39 | ['absolute_humidity_2m:gm3',
'air_density_2m:kgm3', 'ceiling_height_agl:m', 'clear_sky_energy_1h:J',
'clear_sky_rad:W', ...]
('int', ['bool']) : 2 | ['snow_density:kgm3', 'is_estimated']
0.1s = Fit runtime
41 features in original data used to generate 41 features in processed
data.
Train Data (Processed) Memory Usage: 8.02 MB (0.0% of available memory)
Data preprocessing and feature engineering runtime = 0.16s ...
AutoGluon will gauge predictive performance using evaluation metric:
'mean_absolute_error'
This metric's sign has been flipped to adhere to being higher_is_better.
The metric score can be multiplied by -1 to get the metric value.
To change this, specify the eval_metric parameter of Predictor()
use_bag_holdout=True, will use tuning_data as holdout (will not be used for
early stopping).
User-specified model hyperparameters to be fit:
{
    'NN_TORCH': {},
    'KNN': [{'weights': 'uniform', 'ag_args': {'name_suffix': 'Unif'}}],
    'GBM': [{'extra_trees': True, 'feature_fraction': 0.7832570544199176,
'learning_rate': 0.021720607471727896, 'min_data_in_leaf': 3, 'num_leaves': 21,
'ag_args': {'name_suffix': '_r118', 'priority': 17}}],
    'FASTAI': [{'bs': 1024, 'emb_drop': 0.6046989241462619, 'epochs': 48,
'layers': [200, 100, 50], 'lr': 0.00775309042164966, 'ps': 0.09244767444160731,
'ag_args': {'name_suffix': '_r51', 'priority': 12}}],
}
Fitting 4 L1 models ...
Fitting model: KNeighborsUnif_BAG_L1 ...
-19.8149 = Validation score (-mean_absolute_error)
0.03s = Training runtime
0.4s = Validation runtime
Fitting model: NeuralNetTorch_BAG_L1 ...
Fitting 24 child models (S1F1 - S3F8) | Fitting with
ParallelLocalFoldFittingStrategy
-13.3914 = Validation score (-mean_absolute_error)
119.8s = Training runtime
1.08s = Validation runtime
Fitting model: LightGBM_r118_BAG_L1 ...
Fitting 24 child models (S1F1 - S3F8) | Fitting with
ParallelLocalFoldFittingStrategy
-11.2132 = Validation score (-mean_absolute_error)
46.12s = Training runtime
120.54s = Validation runtime
Fitting model: NeuralNetFastAI_r51_BAG_L1 ...
Fitting 24 child models (S1F1 - S3F8) | Fitting with
ParallelLocalFoldFittingStrategy
-12.4891 = Validation score (-mean_absolute_error)

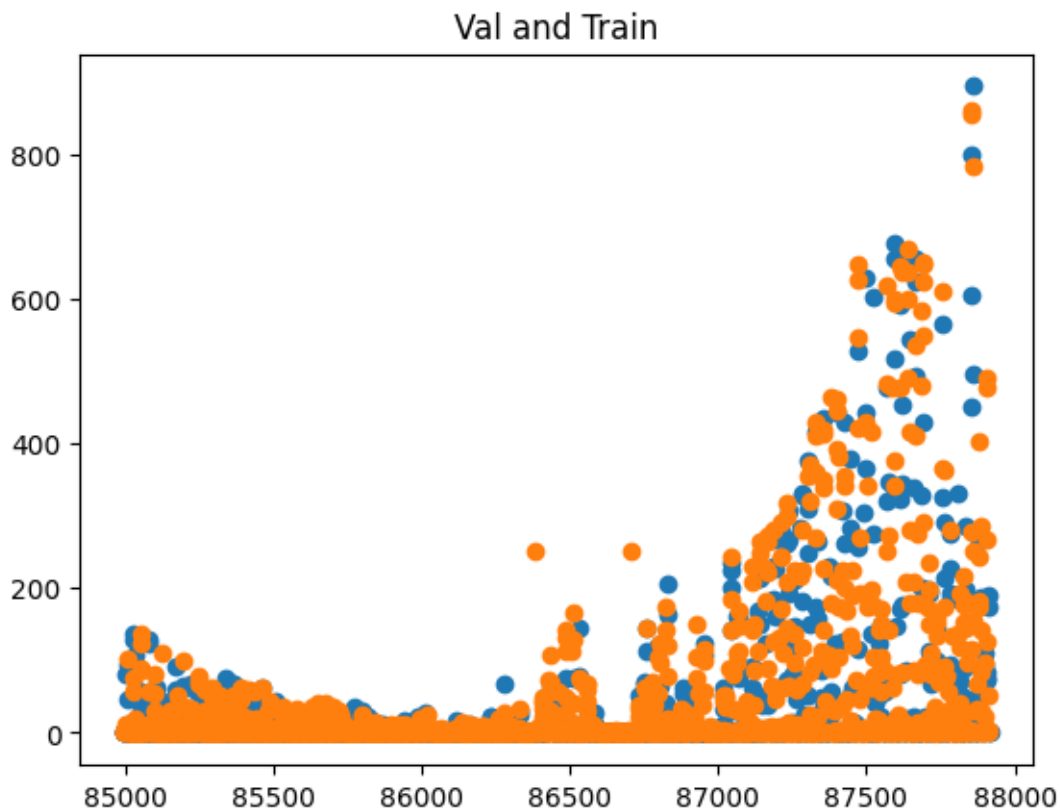
```

```

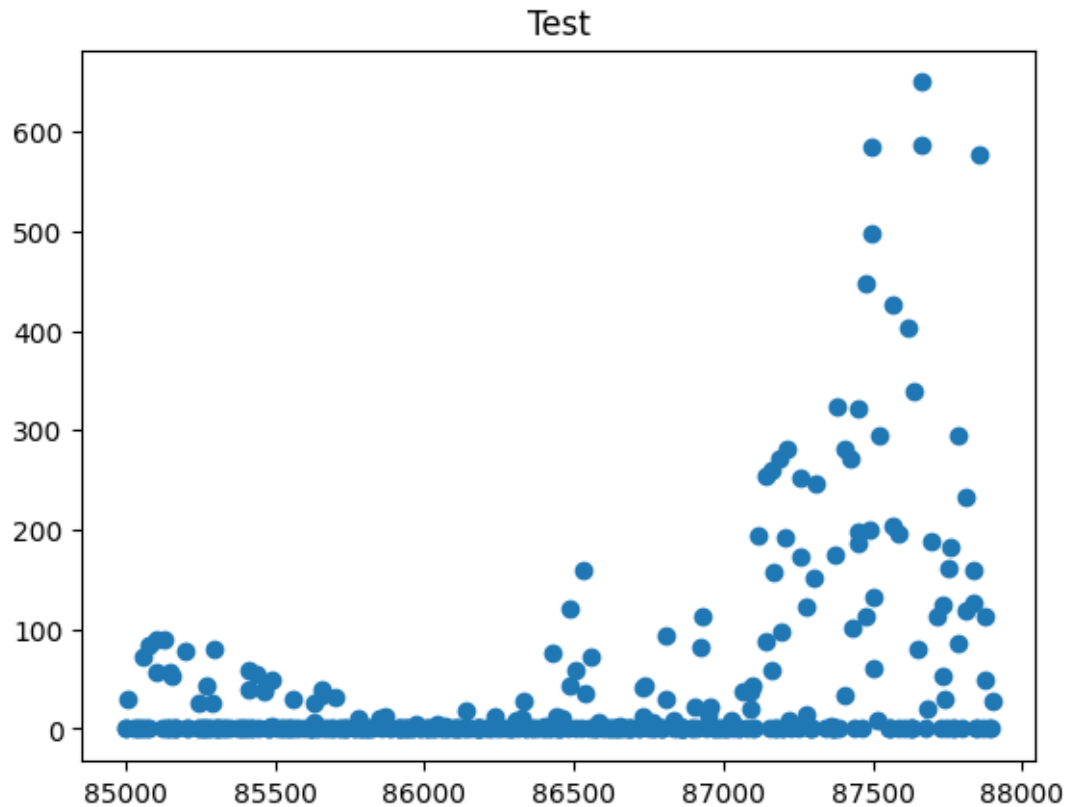
30.01s    = Training    runtime
0.84s     = Validation runtime
Fitting model: WeightedEnsemble_L2 ...
-11.1188      = Validation score    (-mean_absolute_error)
0.17s        = Training    runtime
0.0s         = Validation runtime
AutoGluon training complete, total runtime = 212.66s ... Best model:
"WeightedEnsemble_L2"
TabularPredictor saved. To load, use: predictor =
TabularPredictor.load("AutogluonModels/submission_142_C/")
Evaluation: mean_absolute_error on test data: -11.875635142630587
    Note: Scores are always higher_is_better. This metric score can be
multiplied by -1 to get the metric value.
Evaluations on test data:
{
    "mean_absolute_error": -11.875635142630587,
    "root_mean_squared_error": -28.99915713642264,
    "mean_squared_error": -840.951114622932,
    "r2": 0.9147580094437558,
    "pearsonr": 0.9591648674018699,
    "median_absolute_error": -0.8010042905807495
}

Evaluation on test data:
-11.875635142630587

```



	model	score_test	score_val	pred_time_test
0	WeightedEnsemble_L2	-11.875635	-11.118753	3.301615
122.859288	196.134156	0.003131	0.000628	
0.173273	2	True	5	
1	LightGBM_r118_BAG_L1	-11.977019	-11.213239	2.352773
120.538184	46.121330	2.352773	120.538184	
46.121330	1	True	3	
2	NeuralNetFastAI_r51_BAG_L1	-12.945380	-12.489102	0.377565
0.844247	30.014023	0.377565	0.844247	
30.014023	1	True	4	
3	NeuralNetTorch_BAG_L1	-13.236768	-13.391358	0.546498
1.076832	119.800254	0.546498	1.076832	
119.800254	1	True	2	
4	KNeighborsUnif_BAG_L1	-20.049167	-19.814903	0.021647
0.399398	0.025275	0.021647	0.399398	
0.025275	1	True	1	



```
[ ]: # save leaderboards to csv
pd.concat(leaderboards).to_csv(f"leaderboards/{new_filename}.csv")

for i in range(len(predictors)):
    print(f"Predictor {i}:")
    print(predictors[i].
    ↪info()["model_info"]["WeightedEnsemble_L2"]["children_info"]["S1F1"]["model_weights"])
```

Predictor 0:

```
{'LightGBMXT_BAG_L1': 0.2823529411764706, 'NeuralNetTorch_BAG_L1':
0.35294117647058826, 'LightGBM_r118_BAG_L1': 0.16470588235294117,
'LightGBMLarge_BAG_L1': 0.2}
```

Predictor 1:

```
{'NeuralNetFastAI_BAG_L1': 0.14893617021276595, 'NeuralNetTorch_BAG_L1':
0.2872340425531915, 'LightGBM_r118_BAG_L1': 0.3829787234042553,
'NeuralNetFastAI_r145_BAG_L1': 0.18085106382978725}
```

Predictor 2:

```
{'KNeighborsUnif_BAG_L1': 0.036585365853658534, 'NeuralNetTorch_BAG_L1':
0.012195121951219513, 'LightGBM_r118_BAG_L1': 0.8170731707317073,
'NeuralNetFastAI_r51_BAG_L1': 0.13414634146341464}
```

5 Submit

```
[ ]: import pandas as pd
import matplotlib.pyplot as plt
```

```
future_test_data = TabularDataset('X_test_raw.csv')
future_test_data["ds"] = pd.to_datetime(future_test_data["ds"])
```

Loaded data from: X_test_raw.csv | Columns = 45 / 45 | Rows = 4608 -> 4608

```
[ ]: test_ids = TabularDataset('test.csv')
test_ids["time"] = pd.to_datetime(test_ids["time"])
# merge test_data with test_ids
future_test_data_merged = pd.merge(future_test_data, test_ids, how="inner",
    ↪right_on=["time", "location"], left_on=["ds", "location"])
```

Loaded data from: test.csv | Columns = 4 / 4 | Rows = 2160 -> 2160

```
[ ]: # predict, grouped by location
predictions = []
location_map = {
    "A": 0,
    "B": 1,
    "C": 2
}
for loc, group in future_test_data.groupby('location'):
    i = location_map[loc]
    subset = future_test_data_merged[future_test_data_merged["location"] ==
    ↪loc].reset_index(drop=True)
    pred = predictors[i].predict(subset)
    subset["prediction"] = pred
    predictions.append(subset)

    # get past predictions
    #tuning_data.loc[tuning_data["location"] == loc, "prediction"] =
    ↪predictors[i].predict(tuning_data[tuning_data["location"] == loc])
    if use_test_data:
        # get predictions for local test_data
        test_data.loc[test_data["location"] == loc, "prediction"] =
    ↪predictors[i].predict(test_data[test_data["location"] == loc])
```

```
[ ]: for loc, idx in location_map.items():
    fig, ax = plt.subplots(figsize=(20, 10))
    # plot train data
    train_data[train_data["location"]==loc].plot(x='ds', y='y', ax=ax,
    ↪label="train data")
    tuning_data[tuning_data["location"]==loc].plot(x='ds', y='y', ax=ax,
    ↪label="tune data")
```



```

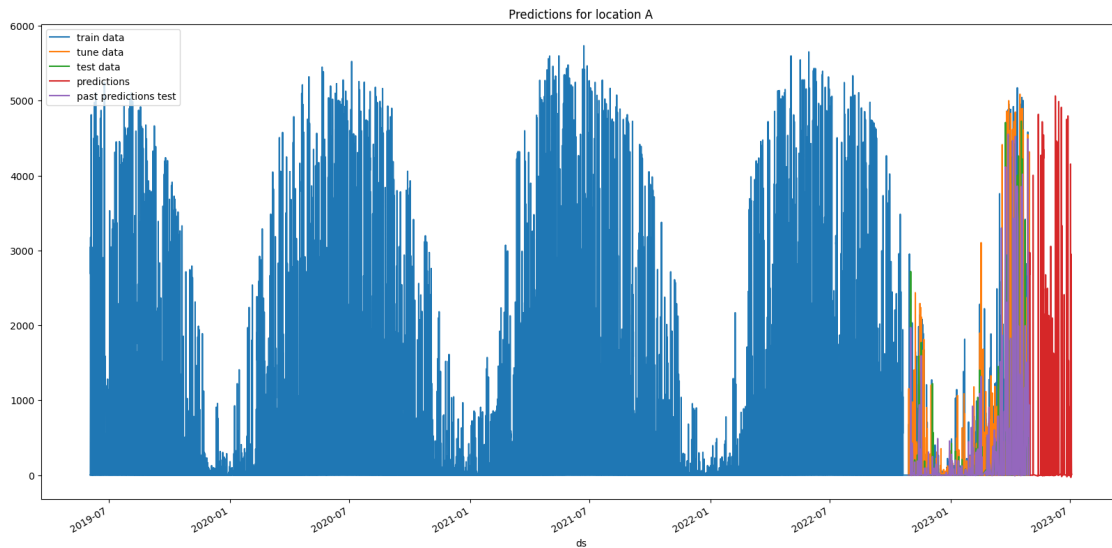
if use_test_data:
    test_data[test_data["location"]==loc].plot(x='ds', y='y', ax=ax,
↳label="test data")

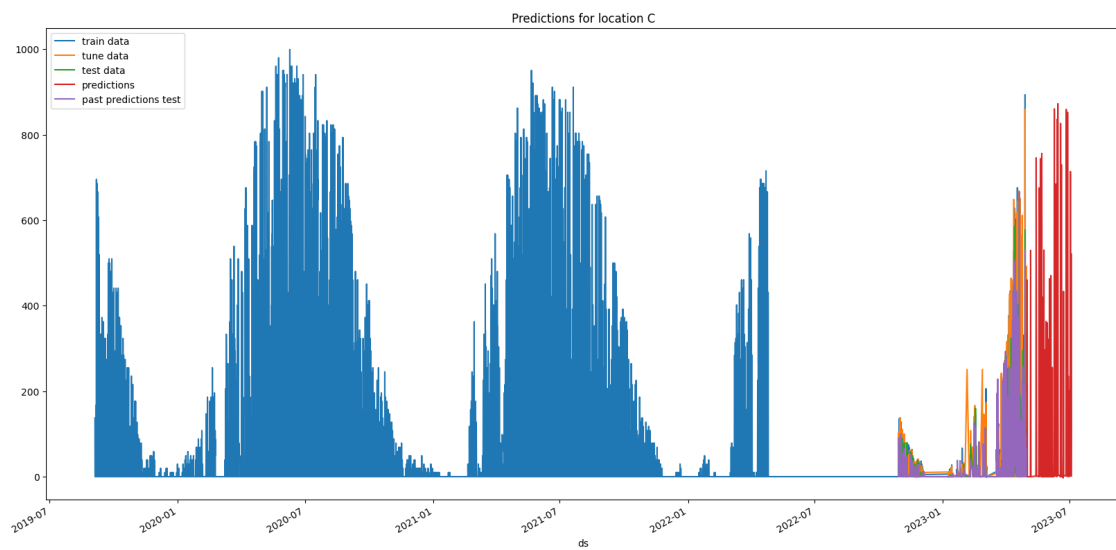
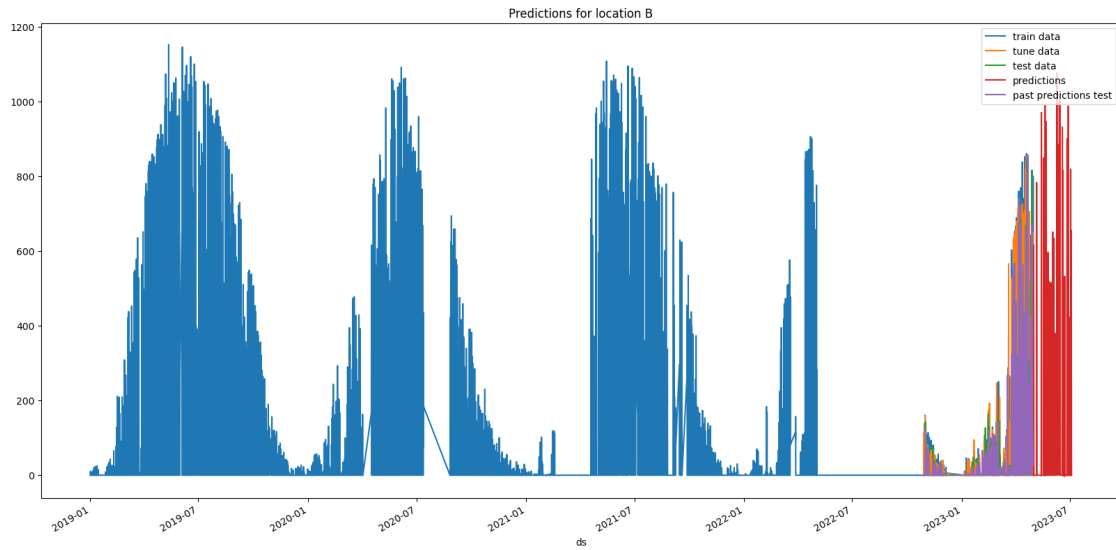
# plot predictions
predictions[idx].plot(x='ds', y='prediction', ax=ax, label="predictions")

# plot past predictions
#tuning_data[tuning_data["location"]==loc].plot(x='ds', y='prediction',
↳ax=ax, label="past predictions tune")
if use_test_data:
    test_data[test_data["location"]==loc].plot(x='ds', y='prediction',
↳ax=ax, label="past predictions test")

ax.set_title(f"Predictions for location {loc}")

```





```
[ ]: temp_predictions = [prediction.copy() for prediction in predictions]
if clip_predictions:
    # clip predictions smaller than 0 to 0
    for pred in temp_predictions:
        # print smallest prediction
        print("Smallest prediction:", pred["prediction"].min())
        pred.loc[pred["prediction"] < 0, "prediction"] = 0
        print("Smallest prediction after clipping:", pred["prediction"].min())
```

```
# concatenate predictions
submissions_df = pd.concat(temp_predictions)
submissions_df = submissions_df[["id", "prediction"]]
submissions_df
```

```
Smallest prediction: -30.471474
Smallest prediction after clipping: 0.0
Smallest prediction: -3.0752246
Smallest prediction after clipping: 0.0
Smallest prediction: -3.1478677
Smallest prediction after clipping: 0.0
```

```
[ ]:      id  prediction
0      0    0.000000
1      1    0.000000
2      2    0.000000
3      3   30.471388
4      4  309.942200
..    ...      ...
715   2155   62.359646
716   2156   35.705055
717   2157    8.107715
718   2158    1.970510
719   2159    1.575655
```

```
[2160 rows x 2 columns]
```

```
[ ]: # Save the submission
print(f"Saving submission to submissions/{new_filename}.csv")
submissions_df.to_csv(os.path.join('submissions', f"{new_filename}.csv"),
    ↪index=False)
```

```
Saving submission to submissions/submission_142.csv
```

```
[ ]: # feature importance
print("\033[1m" + "Calculating feature importance for location A..." +
    ↪"\033[0m")
print(predictors[0].feature_importance(feature_stage="original",
    ↪data=test_data[test_data["location"] == "A"], time_limit=60*10))
print("\033[1m" + "Calculating feature importance for location B..." +
    ↪"\033[0m")
print(predictors[1].feature_importance(feature_stage="original",
    ↪data=test_data[test_data["location"] == "B"], time_limit=60*10))
print("\033[1m" + "Calculating feature importance for location C..." +
    ↪"\033[0m")
print(predictors[2].feature_importance(feature_stage="original",
    ↪data=test_data[test_data["location"] == "C"], time_limit=60*10))
```

These features in provided data are not utilized by the predictor and will be ignored: ['ds', 'elevation:m', 'snow_drift:idx', 'location', 'prediction']
Computing feature importance via permutation shuffling for 41 features using 361 rows with 10 shuffle sets... Time limit: 600s...

Calculating feature importance for location A...

6299.62s = Expected runtime (629.96s per shuffle set)

556.69s = Actual runtime (Completed 8 of 10 shuffle sets) (Early stopping due to lack of time...)

These features in provided data are not utilized by the predictor and will be ignored: ['ds', 'elevation:m', 'location', 'prediction']

Computing feature importance via permutation shuffling for 42 features using 361 rows with 10 shuffle sets... Time limit: 600s...

	importance	stddev	p_value	n \
direct_rad:W	104.069145	5.795764	1.502413e-10	8
clear_sky_rad:W	68.575522	10.068412	1.265379e-07	8
diffuse_rad:W	62.134230	7.578349	3.516900e-08	8
sun_elevation:d	25.859155	5.634662	1.873632e-06	8
sun_azimuth:d	24.264215	10.749928	1.864269e-04	8
clear_sky_energy_1h:J	18.317073	7.712776	1.365413e-04	8
direct_rad_1h:J	16.850952	3.857918	2.614651e-06	8
effective_cloud_cover:p	12.321047	2.588230	1.462885e-06	8
total_cloud_cover:p	12.293495	2.463435	1.062471e-06	8
diffuse_rad_1h:J	9.097221	3.379883	6.246524e-05	8
relative_humidity_1000hPa:p	8.966096	2.066613	2.735902e-06	8
snow_water:kgm2	7.251352	3.935467	6.186022e-04	8
ceiling_height_agl:m	5.720491	2.248042	8.893116e-05	8
wind_speed_10m:ms	5.230880	2.534388	3.192441e-04	8
is_day:idx	4.823968	1.250933	6.017868e-06	8
cloud_base_agl:m	4.515201	1.510970	3.203023e-05	8
is_in_shadow:idx	4.280729	1.574346	5.857360e-05	8
visibility:m	3.775104	2.571516	2.141486e-03	8
sfc_pressure:hPa	3.747542	1.754387	2.600879e-04	8
pressure_100m:hPa	3.722692	2.085865	7.416005e-04	8
precip_type_5min:idx	2.808903	3.550853	3.015316e-02	8
pressure_50m:hPa	2.731133	1.382244	4.128748e-04	8
fresh_snow_12h:cm	2.050780	2.845603	4.045171e-02	8
msl_pressure:hPa	2.022007	1.133079	7.420672e-04	8
t_1000hPa:K	1.856515	1.442572	4.143340e-03	8
super_cooled_liquid_water:kgm2	1.788779	1.977381	1.881276e-02	8
fresh_snow_24h:cm	1.509543	2.095908	4.052787e-02	8
snow_depth:cm	1.365274	1.348776	1.211626e-02	8
precip_5min:mm	1.279686	2.198281	7.182611e-02	8
fresh_snow_6h:cm	0.982078	2.069044	1.106651e-01	8
dew_point_2m:K	0.951454	0.735408	4.038617e-03	8
absolute_humidity_2m:gm3	0.870497	0.518454	1.042967e-03	8
air_density_2m:kgm3	0.665587	0.442231	1.880318e-03	8

rain_water:kgm2	0.375284	0.525337	4.153244e-02	8
snow_density:kgm3	0.278830	1.126102	2.531601e-01	8
snow_melt_10min:mm	0.180914	0.702212	2.449199e-01	8
prob_rime:p	0.094046	0.609714	3.378885e-01	8
is_estimated	-0.000001	0.000000	5.000000e-01	8
dew_or_rime:idx	-0.301856	0.313019	9.852790e-01	8
fresh_snow_1h:cm	-0.549453	1.493392	8.336780e-01	8
fresh_snow_3h:cm	-0.636534	1.803587	8.242912e-01	8

	p99_high	p99_low
direct_rad:W	111.239979	96.898312
clear_sky_rad:W	81.032707	56.118337
diffuse_rad:W	71.510573	52.757886
sun_elevation:d	32.830664	18.887646
sun_azimuth:d	37.564608	10.963822
clear_sky_energy_1h:J	27.859737	8.774409
direct_rad_1h:J	21.624176	12.077727
effective_cloud_cover:p	15.523346	9.118748
total_cloud_cover:p	15.341390	9.245599
diffuse_rad_1h:J	13.278996	4.915447
relative_humidity_1000hPa:p	11.523022	6.409169
snow_water:kgm2	12.120525	2.382179
ceiling_height_agl:m	8.501891	2.939091
wind_speed_10m:ms	8.366562	2.095198
is_day:idx	6.371690	3.276247
cloud_base_agl:m	6.384655	2.645747
is_in_shadow:idx	6.228594	2.332863
visibility:m	6.956724	0.593485
sfc_pressure:hPa	5.918164	1.576919
pressure_100m:hPa	6.303437	1.141947
precip_type_5min:idx	7.202211	-1.584405
pressure_50m:hPa	4.441321	1.020945
fresh_snow_12h:cm	5.571514	-1.469953
msl_pressure:hPa	3.423914	0.620101
t_1000hPa:K	3.641343	0.071686
super_cooled_liquid_water:kgm2	4.235302	-0.657744
fresh_snow_24h:cm	4.102713	-1.083628
snow_depth:cm	3.034053	-0.303504
precip_5min:mm	3.999518	-1.440146
fresh_snow_6h:cm	3.542011	-1.577855
dew_point_2m:K	1.861340	0.041567
absolute_humidity_2m:gm3	1.511955	0.229038
air_density_2m:kgm3	1.212739	0.118434
rain_water:kgm2	1.025259	-0.274692
snow_density:kgm3	1.672105	-1.114445
snow_melt_10min:mm	1.049729	-0.687901
prob_rime:p	0.848417	-0.660325
is_estimated	-0.000001	-0.000001

dew_or_rime:idx	0.085427	-0.689140
fresh_snow_1h:cm	1.298253	-2.397159
fresh_snow_3h:cm	1.594962	-2.868029

Calculating feature importance for location B...

1866.73s = Expected runtime (186.67s per shuffle set)

338.5s = Actual runtime (Completed 10 of 10 shuffle sets)

These features in provided data are not utilized by the predictor and will be ignored: ['ds', 'elevation:m', 'snow_drift:idx', 'location', 'prediction']

Computing feature importance via permutation shuffling for 41 features using 360 rows with 10 shuffle sets... Time limit: 600s...

	importance	stddev	p_value	n \
clear_sky_rad:W	31.587960	2.610076	1.409745e-11	10
direct_rad:W	19.005209	1.239041	1.686452e-12	10
diffuse_rad:W	14.643005	1.777099	4.356866e-10	10
sun_elevation:d	10.150755	1.139892	2.183115e-10	10
clear_sky_energy_1h:J	7.924555	1.267501	5.025189e-09	10
sun_azimuth:d	7.634168	1.078358	1.675814e-09	10
direct_rad_1h:J	3.855799	0.595161	3.671306e-09	10
effective_cloud_cover:p	3.348848	0.568381	8.473403e-09	10
diffuse_rad_1h:J	3.307861	0.845782	2.975928e-07	10
fresh_snow_24h:cm	2.204943	0.612355	6.011762e-07	10
is_in_shadow:idx	1.485250	0.173928	3.170873e-10	10
relative_humidity_1000hPa:p	1.408372	0.381177	4.831119e-07	10
cloud_base_agl:m	1.386407	0.337585	1.958281e-07	10
pressure_50m:hPa	1.277418	0.278627	7.577772e-08	10
snow_water:kgm2	1.238569	0.361304	9.104212e-07	10
total_cloud_cover:p	0.986583	0.265938	4.667283e-07	10
pressure_100m:hPa	0.852863	0.256116	1.161805e-06	10
wind_speed_10m:ms	0.833729	0.194466	1.352852e-07	10
msl_pressure:hPa	0.799104	0.232041	8.759535e-07	10
t_1000hPa:K	0.704985	0.220162	1.611908e-06	10
visibility:m	0.703783	0.218386	1.528165e-06	10
ceiling_height_agl:m	0.690568	0.214369	1.533144e-06	10
super_cooled_liquid_water:kgm2	0.647694	0.370226	1.823310e-04	10
snow_density:kgm3	0.604972	0.518949	2.512409e-03	10
air_density_2m:kgm3	0.534649	0.375315	7.393871e-04	10
is_day:idx	0.509820	0.112740	8.535795e-08	10
sfc_pressure:hPa	0.450195	0.194688	2.252136e-05	10
fresh_snow_12h:cm	0.415491	0.251530	2.732797e-04	10
precip_type_5min:idx	0.371654	0.358296	4.763316e-03	10
fresh_snow_6h:cm	0.302202	0.242729	1.710480e-03	10
rain_water:kgm2	0.301656	0.182073	2.676543e-04	10
absolute_humidity_2m:gm3	0.231145	0.056729	2.095480e-07	10
snow_depth:cm	0.199873	0.137941	6.619777e-04	10
precip_5min:mm	0.189552	0.262498	2.414245e-02	10
dew_point_2m:K	0.177787	0.105088	2.311960e-04	10
dew_or_rime:idx	0.133855	0.124232	3.891740e-03	10

fresh_snow_1h:cm	0.110061	0.096312	2.813352e-03	10
fresh_snow_3h:cm	0.107795	0.112304	7.062264e-03	10
prob_rime:p	0.057843	0.145875	1.207319e-01	10
snow_melt_10min:mm	0.045337	0.206622	2.526413e-01	10
snow_drift:idx	0.000000	0.000000	5.000000e-01	10
is_estimated	0.000000	0.000000	5.000000e-01	10

	p99_high	p99_low
clear_sky_rad:W	34.270305	28.905616
direct_rad:W	20.278557	17.731861
diffuse_rad:W	16.469309	12.816702
sun_elevation:d	11.322208	8.979302
clear_sky_energy_1h:J	9.227150	6.621959
sun_azimuth:d	8.742384	6.525953
direct_rad_1h:J	4.467439	3.244159
effective_cloud_cover:p	3.932966	2.764729
diffuse_rad_1h:J	4.177061	2.438661
fresh_snow_24h:cm	2.834253	1.575633
is_in_shadow:idx	1.663994	1.306507
relative_humidity_1000hPa:p	1.800103	1.016640
cloud_base_agl:m	1.733339	1.039475
pressure_50m:hPa	1.563759	0.991076
snow_water:kgm2	1.609877	0.867260
total_cloud_cover:p	1.259884	0.713282
pressure_100m:hPa	1.116070	0.589656
wind_speed_10m:ms	1.033579	0.633879
msl_pressure:hPa	1.037570	0.560638
t_1000hPa:K	0.931243	0.478727
visibility:m	0.928215	0.479350
ceiling_height_agl:m	0.910872	0.470264
super_cooled_liquid_water:kgm2	1.028171	0.267218
snow_density:kgm3	1.138290	0.071654
air_density_2m:kgm3	0.920356	0.148942
is_day:idx	0.625681	0.393958
sfc_pressure:hPa	0.650273	0.250117
fresh_snow_12h:cm	0.673986	0.156996
precip_type_5min:idx	0.739870	0.003437
fresh_snow_6h:cm	0.551652	0.052753
rain_water:kgm2	0.488770	0.114542
absolute_humidity_2m:gm3	0.289445	0.172846
snow_depth:cm	0.341633	0.058112
precip_5min:mm	0.459318	-0.080214
dew_point_2m:K	0.285785	0.069790
dew_or_rime:idx	0.261527	0.006184
fresh_snow_1h:cm	0.209040	0.011082
fresh_snow_3h:cm	0.223208	-0.007618
prob_rime:p	0.207757	-0.092071
snow_melt_10min:mm	0.257680	-0.167007

```
snow_drift:idx          0.000000  0.000000
is_estimated           0.000000  0.000000
```

Calculating feature importance for location C...

1469.61s = Expected runtime (146.96s per shuffle set)

167.27s = Actual runtime (Completed 10 of 10 shuffle sets)

	importance	stddev	p_value	n \
clear_sky_rad:W	9.492532e+00	0.799178	1.666848e-11	10
clear_sky_energy_1h:J	7.382851e+00	0.668056	3.177582e-11	10
direct_rad:W	5.329617e+00	0.649175	4.500538e-10	10
direct_rad_1h:J	3.410787e+00	0.449938	9.139202e-10	10
sun_elevation:d	3.150492e+00	0.428365	1.195200e-09	10
sun_azimuth:d	3.138013e+00	0.618942	3.159672e-08	10
diffuse_rad:W	2.284770e+00	0.412072	1.444585e-08	10
diffuse_rad_1h:J	1.737366e+00	0.211993	4.571480e-10	10
t_1000hPa:K	1.443522e+00	0.944473	4.648883e-04	10
air_density_2m:kgm3	1.236523e+00	0.564241	3.417133e-05	10
effective_cloud_cover:p	1.085109e+00	0.249225	1.184326e-07	10
is_day:idx	1.073634e+00	0.118328	1.847534e-10	10
fresh_snow_24h:cm	1.069518e+00	0.364482	3.322601e-06	10
is_in_shadow:idx	9.497877e-01	0.162336	9.011840e-09	10
total_cloud_cover:p	8.506052e-01	0.290276	3.360237e-06	10
cloud_base_agl:m	5.502705e-01	0.408595	1.057501e-03	10
snow_water:kgm2	5.149236e-01	0.240466	4.081609e-05	10
dew_point_2m:K	4.391927e-01	0.189333	2.197376e-05	10
relative_humidity_1000hPa:p	3.931047e-01	0.469069	1.323475e-02	10
ceiling_height_agl:m	3.908815e-01	0.192129	6.021803e-05	10
visibility:m	3.753250e-01	0.203091	1.227820e-04	10
snow_density:kgm3	3.222942e-01	0.157444	5.748478e-05	10
precip_type_5min:idx	1.828257e-01	0.157011	2.529091e-03	10
absolute_humidity_2m:gm3	1.621643e-01	0.120629	1.069484e-03	10
super_cooled_liquid_water:kgm2	1.245658e-01	0.087702	7.536740e-04	10
precip_5min:mm	1.113224e-01	0.217238	6.978912e-02	10
rain_water:kgm2	5.352307e-02	0.088592	4.419605e-02	10
pressure_50m:hPa	4.990453e-02	0.203321	2.287793e-01	10
sfc_pressure:hPa	3.411336e-02	0.168589	2.691001e-01	10
pressure_100m:hPa	2.526476e-02	0.221418	3.632755e-01	10
fresh_snow_1h:cm	1.675500e-02	0.015397	3.687800e-03	10
dew_or_rime:idx	1.623237e-02	0.034804	8.717662e-02	10
prob_rime:p	1.583810e-02	0.020512	1.862899e-02	10
msl_pressure:hPa	1.510474e-02	0.163634	3.884884e-01	10
snow_melt_10min:mm	1.352891e-02	0.010079	1.079957e-03	10
snow_depth:cm	1.872858e-04	0.006698	4.657394e-01	10
is_estimated	1.196232e-08	0.000000	5.000000e-01	10
fresh_snow_3h:cm	-2.301741e-04	0.026277	5.107472e-01	10
fresh_snow_12h:cm	-3.753542e-04	0.169013	5.027251e-01	10
fresh_snow_6h:cm	-2.073926e-02	0.101690	7.324694e-01	10
wind_speed_10m:ms	-9.447763e-02	0.194232	9.208077e-01	10

	p99_high	p99_low
clear_sky_rad:W	1.031384e+01	8.671226e+00
clear_sky_energy_1h:J	8.069405e+00	6.696298e+00
direct_rad:W	5.996766e+00	4.662467e+00
direct_rad_1h:J	3.873183e+00	2.948392e+00
sun_elevation:d	3.590717e+00	2.710266e+00
sun_azimuth:d	3.774093e+00	2.501934e+00
diffuse_rad:W	2.708252e+00	1.861288e+00
diffuse_rad_1h:J	1.955228e+00	1.519503e+00
t_1000hPa:K	2.414146e+00	4.728979e-01
air_density_2m:kgm3	1.816387e+00	6.566587e-01
effective_cloud_cover:p	1.341235e+00	8.289834e-01
is_day:idx	1.195238e+00	9.520303e-01
fresh_snow_24h:cm	1.444091e+00	6.949443e-01
is_in_shadow:idx	1.116618e+00	7.829572e-01
total_cloud_cover:p	1.148918e+00	5.522924e-01
cloud_base_agl:m	9.701787e-01	1.303623e-01
snow_water:kgm2	7.620476e-01	2.677995e-01
dew_point_2m:K	6.337679e-01	2.446175e-01
relative_humidity_1000hPa:p	8.751617e-01	-8.895233e-02
ceiling_height_agl:m	5.883305e-01	1.934324e-01
visibility:m	5.840395e-01	1.666105e-01
snow_density:kgm3	4.840976e-01	1.604908e-01
precip_type_5min:idx	3.441839e-01	2.146741e-02
absolute_humidity_2m:gm3	2.861337e-01	3.819500e-02
super_cooled_liquid_water:kgm2	2.146965e-01	3.443513e-02
precip_5min:mm	3.345756e-01	-1.119307e-01
rain_water:kgm2	1.445683e-01	-3.752220e-02
pressure_50m:hPa	2.588549e-01	-1.590458e-01
sfc_pressure:hPa	2.073705e-01	-1.391438e-01
pressure_100m:hPa	2.528134e-01	-2.022838e-01
fresh_snow_1h:cm	3.257790e-02	9.321038e-04
dew_or_rime:idx	5.200031e-02	-1.953558e-02
prob_rime:p	3.691781e-02	-5.241607e-03
msl_pressure:hPa	1.832693e-01	-1.530598e-01
snow_melt_10min:mm	2.388743e-02	3.170387e-03
snow_depth:cm	7.070848e-03	-6.696276e-03
is_estimated	1.196232e-08	1.196232e-08
fresh_snow_3h:cm	2.677395e-02	-2.723430e-02
fresh_snow_12h:cm	1.733169e-01	-1.740676e-01
fresh_snow_6h:cm	8.376608e-02	-1.252446e-01
wind_speed_10m:ms	1.051322e-01	-2.940875e-01

```
[ ]: # save this notebook to submissions folder
import subprocess
import os
```

```
subprocess.run(["jupyter", "nbconvert", "--to", "pdf", "--output", os.path.  
↳join('notebook_pdfs', f"{new_filename}.pdf"), "short_1.ipynb"])
```

```
[NbConvertApp] Converting notebook short_1.ipynb to pdf  
/opt/conda/lib/python3.10/site-packages/nbconvert/utils/pandoc.py:51:  
RuntimeWarning: You are using an unsupported version of pandoc (2.9.2.1).  
Your version must be at least (2.14.2) but less than (4.0.0).  
Refer to https://pandoc.org/installing.html.  
Continuing with doubts...  
    check_pandoc_version()  
[NbConvertApp] Support files will be in notebook_pdfs/submission_142_files/  
[NbConvertApp] Making directory  
./notebook_pdfs/submission_142_files/notebook_pdfs  
[NbConvertApp] Writing 162229 bytes to notebook.tex  
[NbConvertApp] Building PDF  
[NbConvertApp] Running xelatex 3 times: ['xelatex', 'notebook.tex', '-quiet']  
[NbConvertApp] Running bibtex 1 time: ['bibtex', 'notebook']  
[NbConvertApp] WARNING | bibtex had problems, most likely because there were no  
citations  
[NbConvertApp] PDF successfully created  
[NbConvertApp] Writing 534855 bytes to notebook_pdfs/submission_142.pdf  
[ ]: CompletedProcess(args=['jupyter', 'nbconvert', '--to', 'pdf', '--output',  
    'notebook_pdfs/submission_142.pdf', 'short_1.ipynb'], returncode=0)
```